



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ  
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT  
INSTITUTE OF INFORMATICS

# ANALÝZA A NÁVRH ELEKTRONICKÉHO OBCHODU

ANALYSIS AND DESIGN OF ELECTRONIC COMMERCE

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. DAVID STRÁNSKÝ

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. ZUZANA NĚMCOVÁ, Ph.D.

BRNO 2013

# **ZADÁNÍ DIPLOMOVÉ PRÁCE**

**Stránský David, Bc.**

---

Informační management (6209T015)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává diplomovou práci s názvem:

**Analýza a návrh elektronického obchodu**

v anglickém jazyce:

**Analysis and Design of Electronic Commerce**

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení

Seznam použité literatury

Závěr

Přílohy

Seznam odborné literatury:

BLAŽKOVÁ, M. Jak využít internet v marketingu, 1. vyd. Praha: Grada, 2005. 156 s. ISBN 80-247-1095-1.

DVOŘÁK, J. Elektronický obchod, Brno, 2004. 78 s.

FRIMEL, M. Elektronický obchod, 1. vyd. Praha: Prospektrum, 2002. 321 s. ISBN 80-7175-114-6.

KRUG, S. Webdesign - Nenutíte uživatele přemýšlet. 2. vyd. Brno: Computer Press, 2006. 168 s. ISBN 80-7226-892-9.

SMIČKA, R. Optimalizace pro vyhledávače - SEO. Dubany: Jaroslava Smičková, 2004. 120 s. ISBN 80-239-2961-5.

Vedoucí diplomové práce: Ing. Zuzana Němcová, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2012/2013.

L.S.

---

doc. RNDr. Bedřich Půža, CSc.  
Ředitel ústavu

---

doc. Ing. et Ing. Stanislav Škapa, Ph.D.  
Děkan fakulty

V Brně, dne 21.05.2013

## **Abstrakt**

Diplomová práce je zaměřena na analýzu, návrh a implementaci internetového obchodu. První část je věnovaná internetovému marketingu, optimalizaci pro vyhledavače a modelování elektronického obchodu z teoretického hlediska. Analytická část je zaměřena na popis běžné funkcionality internetového obchodu a seznámení s některými dostupnými řešeními a kroky při jeho realizaci. Na základě teoretických a analytických poznatků bude navržen a implementován model internetového obchodu, který poslouží jako výchozí část pro budoucí aplikaci na míru konkrétního zákazníka.

## **Abstract**

This master's thesis focuses on the analysis, design and implementation of Electronic Commerce. The first part is devoted to Internet marketing, search engine optimization and modeling of Electronic Commerce from a theoretical point of view. The analytical part focuses on the description of the common e-commerce functionalities and meeting with some available solutions and steps for its implementation. There will be designed and implemented model of E-commerce based on theoretical and analytical knowledge. This model will serve as the basic part for future application, which fits specific customer's needs.

## **Klíčová slova**

internetový marketing, optimalizace pro vyhledavače, elektronické obchodování, internetový obchod, B2C, UML, PHP, MySQL, MVC, Yii, webový server

## **Key words**

internet marketing, search engine optimization, e-commerce, e-business, B2C, UML, PHP, MySQL, MVC, Yii, web server

### **Bibliografická citace VŠKP dle ČSN ISO 690**

STRÁNSKÝ, D. *Analýza a návrh elektronického obchodu*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2013. 85 s. Vedoucí diplomové práce Ing. Zuzana Němcová, Ph.D.

## **Prohlášení**

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně.  
Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 20.5.2013

.....

podpis studenta

## **Poděkování**

Na tomto místě bych chtěl poděkovat své vedoucí Ing. Zuzaně Němcové, Ph.D. za odborné rady a připomínky při psaní mé diplomové práce.

# OBSAH

ÚVOD.....	9
1 VYMEZENÍ PROBLÉMU A CÍLE PRÁCE.....	10
2 TEORETICKÁ VÝCHODISKA PRÁCE.....	11
2.1 Internetový marketing.....	11
2.1.1 Internetový marketing jako proces.....	12
2.1.2 Začlenění internetu do firemního marketingového mixu.....	13
2.1.3 Internetový marketingový mix 4C.....	14
2.1.4 Komunikační mix na internetu.....	16
2.1.5 Nákupní cyklus na internetu.....	16
2.2 SEO - optimalizace pro vyhledávače.....	20
2.2.1 Co je SEO?.....	20
2.2.2 Jak uživatelé vyhledávají.....	21
2.2.3 Jaké jsou cíle SEO.....	23
2.2.4 On-page faktory SEO při realizaci webu.....	23
2.2.5 Offpage faktory – linkbuilding.....	30
2.3 Elektronický obchod.....	33
2.3.1 Systém a model elektronického obchodu.....	33
2.3.2 Druhy elektronického obchodování.....	37
3 ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE.....	40
3.1 Běžná funkcionality e-shopu.....	40
3.1.1 Frontend.....	40
3.1.2 Backend.....	43
3.2 Dostupné možnosti při realizaci e-shopu.....	47
3.2.1 Open source řešení.....	47
3.2.2 Obchod na míru.....	52
3.3 Analýza současné situace firmy.....	54
3.3.1 Popis firmy.....	54
3.3.2 SWOT analýza.....	54
4 NÁVRH ŘEŠENÍ.....	57
4.1 Požadavky na aplikaci.....	57
4.2 Návrh aplikace.....	58
4.2.1 Aplikace internetového obchodu a uživatelé.....	58
4.2.2 Frontend - případy užití.....	58
4.2.3 Backend – případy užití.....	59
4.2.4 Popis procesu objednání zboží.....	60
4.2.5 Návrh databázového modelu.....	62
4.3 Implementace aplikace.....	68
4.3.1 Použité technologie.....	68
4.3.2 Architektura aplikace - MVC.....	70
4.3.3 Adresářová struktura a nastavení.....	71
4.3.4 Programování.....	74
4.3.5 Vzhled aplikace.....	77
ZÁVĚR.....	78
POUŽITÉ ZDROJE.....	79
SEZNAM OBRÁZKŮ.....	82
SEZNAM ZKRATEK.....	84
SEZNAM PŘÍLOH.....	85



## ÚVOD

V teoretické části diplomové práce, zabývající se internetovým marketingem se budu snažit popsat dostupné techniky komunikace se zákazníkem a proč si zvolit internet jako cílové médium. Kapitola o nákupním cyklu nám řekne, jak se dotyčný klient rozhoduje při hledání a výběru zboží. Do internetového marketingu zapadá celá řada činností a oborů, popsat jen detailněji jeden konkrétní by vydalo na několik knih, proto se budu snažit vybrat podle mě důležité informace a kroky pro jednu disciplínu - optimalizace pro vyhledavače. Nakonec popíšu teoreticky elektronický obchod jako model a také základní typy elektronického podnikání.

Podstatou analytické části bude popsat, zhodnotit a na závěr vybrat vhodnou metodu řešení pro realizaci nového internetového obchodu. První část analýzy si klade za cíl seznámit čtenáře se základní funkcionalitou většiny online obchodů. Následně popíšeme dostupné metody implementace takovýchto řešení a přibližnou časovou náročnost na realizaci. Nakonec provedu SWOT analýzu obchodu zákazníka, na jejíž základě rozhodnu o dalších krocích.

Praktická část se bude věnovat návrhu a programování internetového obchodu. Nejprve budou specifikovány případy užití aplikace, ilustrativně popíšu jeden vybraný firemní proces a navrhnu kompletní databázový model, na jehož základě vygeneruji databázovou vrstvu aplikace. Také popíšu jednotlivé entity tohoto modelu a vzájemné vazby mezi nimi.

Před vlastním programování internetového obchodu vyberu vhodné technologie pro jeho implementaci. Proces programování je asi nejsložitější část celé práce a klade důraz na dobrý předcházející návrh a správné techniky tvorby aplikací, budu se je tedy snažit dodržovat a udělat internetový obchod do budoucna snadno rozšiřitelný a hlavně prakticky uplatnitelný.

V odevzdání práce bude zahrnuta naprogramovaná aplikace základního modelu internetového obchodu. Použité materiály budou citovány podle platných norem.

# 1 VYMEZENÍ PROBLÉMU A CÍLE PRÁCE

## Vymezení problému

Byl jsem osloven zákazníkem v záležitosti implementace nového internetového obchodu. V této diplomové práci se budu zabývat výběrem vhodného řešení a jeho následnou realizací. Analýze možností bude přecházet teoretická řešení týkající se elektronického obchodování a internetového marketingu.

## Cíle práce

Cílem diplomové práce je **vytvoření modelu internetového obchodu**, který bude co možná nejefektivněji a harmonicky pracovat. Cíle je dosaženo tím, že jednotlivé podsystémy vzájemně kooperují. Toho bude dosaženo jak objektovým programováním, tak i správnými praktikami psaní programového kódu tzv. „*best practices*“. Výsledkem bude snadno spravovatelný projekt, který bude otevřen budoucím rozšířením.

## Použité metody

Pro účely vyhodnocení internetového obchodu firmy bude v analytické části použita metoda SWOT.

V návrhové části použiji pro definici chování systému UML Use case diagramy. K ilustrativnímu popisu procesu aplikuji znalost tvorby EPC diagramu a slovní popis. E-R diagramy se používají ke specifikaci databázového modelu, ani tato práce nebude výjimkou a osvojenou metodu využiji. Při tvorbě cílové aplikace využiji znalosti techniky objektového programování.

## 2 TEORETICKÁ VÝCHODISKA PRÁCE

### 2.1 Internetový marketing

*„Marketing je proces zjišťování potřeb a požadavků zákazníků za účelem poskytnout jim takový produkt, který pro ně bude představovat přidanou hodnotu, a firmě přinese zisk.“ (1, str. 17)*

Výše zmíněná citace platí beze zbytku i pro internetové prostředí. Internet jako médium s masovým záběrem, poskytující nám nepřeberné množství možností oslovení cílového zákazníka – reklama ve vyhledávačích (placená, neplacená), bannerová reklama až po využití sociálních sítí či rozesílání nabídek přes elektronickou poštu. Tato výhoda ale může být zároveň omezením, jelikož zásah je pouze na uživatele, kteří mají přístup k internetovému připojení. Je tedy vhodné kombinovat offline reklamu (billboardy, letáky, plakáty, televizní reklama) s možnostmi internetu a vytvořit tak firemní image, kde jsou tyto prostředky ve vzájemné synergii.

Oproti klasickým formám marketingu je internetový marketing zaměřen více na potřeby zákazníka a najdeme ještě více rozdílů (2, str. 31):

- Prostor a čas
- Směr komunikace
- Vztah mezi textem a obrazem
- Interakce
- Nákladů a zdrojů

U offline reklamy platíme za **místo** nebo **čas** pro reklamu nemalé finanční prostředky a tato reklama nemusí přinést žádné zákazníky nebo jen mizivé procento – např. když běží v televizi reklama na jídlo pro domácí mazlíčky, tak pro kolik procent diváků je televizní spot relevantní a stanou se zákazníkem? U internetové reklamy je **směr komunikace** obousměrný – zákazník dostane to, co zrovna hledá a měli bychom se mu snažit co nejvíce vyhovět. HTML jazyk pro web nám dává možnost doplňovat stránky podobnými a doplňujícími informacemi a poté vše **vzájemně provázat** odkazy, doplnit obrázky, video záznamy nebo animacemi. K **interakci** mezi uživatelem a webovou aplikací oproti klasické reklamě, kde konverze může trvat delší čas, dochází ihned.

Návštěvník může objednat zboží online, vyplnit poptávkový, dotazovací formulář, informovat se na zboží v sociálních sítích nebo srovnávacích cen a očekává výsledek co nejdříve. Důsledkem pružnosti internetových technologií je otázka aktualizace firemní prezentace nebo obchodu otázkou editace a uložení dat, nemusí se tedy pokaždé tisknout nové brožury, firemní prospekty a šetříme tak **náklady** firmy (2, str. 31-32).

### **2.1.1 Internetový marketing jako proces**

Při realizaci naší strategie si můžeme celý proces zjednodušit na jednotlivé taktické kroky, kterých bychom se měli držet a uplatňovat je na našem konkrétním projektu. Každý jednotlivý segment této činnosti by měl odrážet celkové cíle podniku nebo být jejich podmnožinou. Kroky internetového marketingu jsou (1, str. 21):

#### **Rozhodování**

Před vlastním marketingovým plánem bychom měli vědět zejména pro jaké zákazníky jsou naše služby nebo produkty určeny, tzn. znát naši cílovou skupinu, která může být demograficky, věkově nebo jinak, např. žánrově, odlišena. Cílů webu může být několik - odeslání poptávky, nákup produktu nebo registrace zákazníka, případně čas strávený na stránkách. Cíle jsou ale nejčastěji vyjadřované ziskem, který by nám měl web přinést nebo daným počtem objednávek.

#### **Přidělení kapacit**

Je třeba vyhradit peněžní rozpočet, který ale může být pro každý typ projektu jiný. Vyčlenit ve firmě zaměstnance, který se bude o reklamu a vztahy s veřejností starat nebo najmout externí firmu, která dodá jednotlivé formy marketingu formou subdodávky.

#### **Realizace**

Zde se jedná hlavně o místo a kanály, které budeme k účelu propagace používat. Marketingové nástroje a techniky realizujeme na vlastní aplikaci (e-shopu, webových stránkách, sociálním profilu) i na jiných webech, ze kterých zákazníci přejdou na naši hlavní prezentaci nebo lze také zaplatit vyhledavačům za přivedené návštěvníky pomocí

PPC reklamy či přirozeného vyhledávání. Na svých stránkách můžeme používat konkrétní metody např. doplňkový, křížový prodej, sdílení aktualit se zákazníkem, zboží v akci apod.

### **Monitorování a měření**

U webové návštěvnosti jsou výsledky – pokud máme na svých stránkách umístěn měřicí kód, přímo měřitelné. Lze zaznamenávat návštěvnost, zdroje návštěvnosti – např. placené a neplacené, návštěvnost ze sociálních sítí, z bannerové reklamy či přímou návštěvnost, klíčové slovo, které přivedlo zákazníka nebo konkrétní kampaň.

### **Analýza a vylepšování**

Pokud máme správně naměřená data, je možné je podle předem určených metrik, které jsme si zvolili při rozhodování, vyhodnotit. Můžeme sledovat výnosy a ceny objednávek, lze je dále segmentovat podle zdroje, geografického místa či IP adresy.

## **2.1.2 Začlenění internetu do firemního marketingového mixu**

Všechny firemní aktivity na internetu musí být sladěny s aktivitami v offline prostředí, týká se to např. jednotného firemního vzhledu a sladění grafického designu webových stránek s letáky, vizitkami, atd. nebo vystupováním firmy navenek a její cenovou politikou. Aspekty této problematiky řeší marketingový mix 4P, který je seskupením nástrojů vyjadřujících vztah podniku k jeho okolí a jejichž pomocí podnik uskutečňuje svoje cíle (2, str. 66-67).

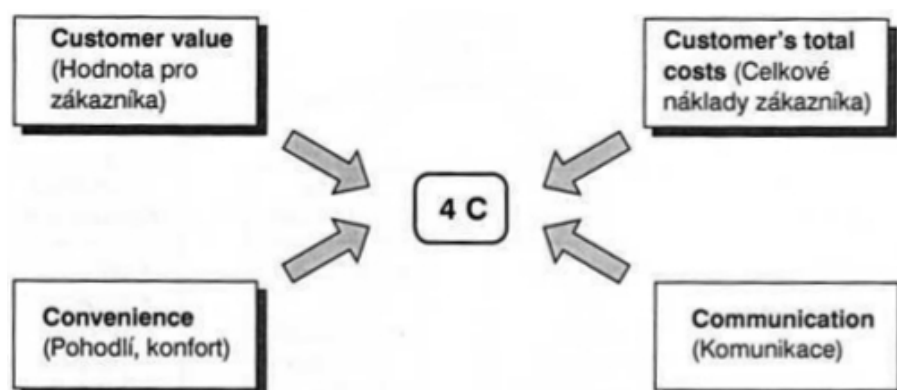
Prvky marketingového mixu 4P jsou (2, str. 67):

- product (výrobek) – zde se rozumí kvalita, design, funkčnost, sortiment
- price (cena) – cena, rabaty, platební podmínky, slevy
- place (distribuce) – možnosti distribučních kanálů, počet přeprodejců, místo prodeje
- promotion (komunikace) – reklama, jednání a vztaky se zákazníky, propagace

### 2.1.3 Internetový marketingový mix 4C

Marketingový mix 4P, je podle mnohých zastaralý. Nezohledňuje totiž zákaznickovy potřeby, proto se častěji používá marketingový mix 4C (3).

Jelikož tento přístup je více zákaznický orientovaný, jsou v něm oproti 4P metodice drobné rozdíly, hodnota pro zákazníka je v předchozím modelu produkt, celkové náklady jsou cena, pohodlí můžeme dát do kontextu s distribucí a komunikace je schodná (3). Tyto prostředky si dále více popíšeme (2, str. 67-70):



Obrázek 1: Marketingový mix 4C na internetu  
Zdroj: (2, str. 68)

#### Hodnota pro zákazníka

Chceme-li působit na internetu, musíme se starat o zákazníka jako o oko v hlavě, tj. musíme zjistit jeho potřeby – vyřčené, skutečné, nevyřčené, potřeby potěšení i tajné potřeby, zákaznickovy preference i chování. Nemenší význam má správná práce s těmito informacemi. Je třeba přistupovat k problému z opačného konce, než bylo léta zaběhlou praxí – nesnažit se prodat zákazníkovi to, co máme zrovna vyrobené a na skladě, ale dodat mu naopak to, co si žádá a vzbuzuje jeho potřebu. Tato nabídka by měla být ušitá na míru konkrétního zákazníka – tzv. personalizace, diferencovat výrobky rozdíly – fyzickými, cenovými, speciálním servisem, či dostupností (2, str. 69).

#### Celkové náklady zákazníka

Zákazník, nakupující na internetu se ne vždy zajímá o konkrétní cenu výrobku, ale spíše o celkové náklady, které jsou spojeny se získáním výrobku – může se jednat o započítané poštovné či náklady na dopravu zboží přepravní firmou, až po zvláštní

přidanou cenu při transakci platební kartou. Můžeme tak do hry zapojit slevové akce, jako sleva na zboží od určité hodnoty objednávky, poštovné zdarma či slevové kupony. Do celkových nákladů na zboží se musí samozřejmě započítat náklady na užívání výrobku a jeho (ekologickou) likvidaci. Od absolutních nákladů si naproti tomu zákazník může pomyslně odečíst čas přepočítaný na peníze, který ušetří při nákupu na internetu, nemusí chodit do kamenného obchodu a stát frontu na zboží, které lze jednoduše vybírat z domu (2, str. 69-70).

### **Pohodlí**

Základem při obchodování či nabízení služeb přes internet je schopnost tohoto média odstranit, při adekvátních nákladech, bariéry v dostupnosti a zprostředkování jeho prodeje. Člověk, který nakupuje jeden den v našem obchodě nebo poptává službu, může zítra nakupovat jinde, jelikož přechod ke konkurenci je přece tak snadný, stačí jen zadat adresu konkurence do prohlížeče. Měli bychom tudíž věnovat snahu tomu, aby se u nás zákazník cítil spokojen, neklást mu zbytečné překážky při procházení našich produktů, stránek, poskytnout mu dostatek informací před konečným nákupem. Naše služby bychom měli podepřít záručním servisem, možnostmi zákaznických dotazů k používání produktu (2, str. 70).

### **Komunikace**

Komunikace se zákazníkem jako součást marketingového mixu zaznamenává největší pokrok a rozšíření. Firma by měla věnovat tomuto prostředku nemalou pozornost, jelikož stále více nabývá na důležitosti. Komunikace se z jednosměrné přeměnila na obousměrnou a zákazník se stále více snaží zapojit do diskuse, aktivně hledat informace o produktech, může např. na sociálních službách pomluvit či pochválit naši službu, šířit dobré jméno firmy. Díky webové analytice můžeme zákazníkovi na základě analýzy hledaných slov poskytnout výrobek přesně na míru nebo udělat balíček služeb (2, str. 70).

Od doby, co je internet tak masově rozšířený, je komunikace, budování dobrého jména firmy a zviditelňování se otázkou přežití online podnikání firmy a obchodu. Budu se tedy zabývat jednotlivými složkami komunikace v dalších kapitolách.

## 2.1.4 Komunikační mix na internetu

Internetový komunikační mix se velice podobá klasickému komunikačnímu mixu s tím rozdílem, že se zde nevyskytuje osobní prodej. Naproti tomu, ale můžeme vzít v potaz složku virální marketing.

Složky mixu jsou (2, str. 82):

- public relations – novinky, zprávy, články, vlastní stránky
- podpora prodeje – partnerské programy (affiliate), slevové akce, kupóny, vzorky
- přímý marketing – emailing, newsletter, on-line chatování
- virální marketing – emaily, videa, vtipy
- reklama – optimalizace pro vyhledavače, reklama ve vyhledávačích, katalogy, specializované vyhledavače, plošná reklama, atd. (4)

Je třeba zmínit, že některé zdroje se v tomto členění v detailech rozcházejí. Např. podle Janoucha (1, str. 26) události nahrazují virální marketing a tento je poté začleněn pod public relations. Zmiňovaná rozdělení nemusí být závazná, vždyť některé prostředky mohou být použity pro podporu prodeje i k realizaci public relations. V dalších kapitolách se budeme zaměřovat hlavně na prostředky reklamy a u internetového obchodu na podporu prodeje.

## 2.1.5 Nákupní cyklus na internetu

Nákupní cyklus nebo rozhodovací proces kupujícího, jak je uváděno v některé literatuře, popisuje chování osoby, která postupně od uvědomění si potřeb postupuje k jejich uspokojení a realizaci (5, str. 336).

### Nákupní rozhodování

Nákupní rozhodování se sestává v podstatě z pěti fází, viz. **obrázek**. Nákupní cyklus ale není jen konečný nákup, je to několik podprocesů před i po samotném nákupu. Marketéři by se tedy měli zaměřit na všechny části nákupního cyklu. Je jasné, že u některých běžných nákupů zákazník některou část cyklu přeskočí nebo změní pořadí (5, str. 336).





Obrázek 2: Nákupní rozhodovací proces  
Zdroj: (5, str. 337)

## Rozpoznání potřeby

První fází cyklu je rozpoznání potřeby. Člověk rozezná svoji potřebu ve chvíli, kdy je schopný pocítit rozdíl mezi reálným a vytouženým stavem. Vnitřní podnět může také v určitých případech vyvolat potřebu. Poté se snaží pomocí známých prostředků tuto potřebu uspokojit. Marketéři by se měli zajímat o to co, kde a jak tyto rozpoznání potřeby spustilo a dovedlo je k nákupu konkrétního produktu (5, str. 337).

U člověka, který potřebu zatím nemá nebo o ní ještě neví, ji lze vyvolat nějakou reklamou – např. televizní spot na novou značku auta, billboard, letáky. V internetovém marketingu se pro tento účel může hodit bannerová či textová reklama v obsahové síti vyhledavačů, zejména pokud uvádíme na trh nový a neznámý produkt.

## Vyhledávání informací

Zákazník si uvědomí svoji potřebu a poté následuje fáze vyhledávání informací. Pokud má nablízku produkt, který podle jeho přání ukojí potřeby a je již rozhodnut si ho pořídit, přeskočí rovnou do fáze nákupu. V opačném případě se bude snažit hledat více informací, které mu poskytnou další vodítka. Kupující se dostane do stadia zvýšené pozornosti a bude získávat informace z různých zdrojů (5, str. 338):

- osobní - rodina, přátelé, známí (mají asi největší váhu)
- komerční – reklamy, plakáty, internet
- veřejné – televize, rádio
- zkušenosti – vlastní zkušenost s užíváním výrobku, služby

Prostřednictvím internetu se dají informace velice snadno vyhledávat pomocí fulltextových vyhledavačů či katalogů. V této fázi se zákazník snaží spíše najít recenze typu „Jak vybrat dobré auto (počítač)?“. Marketér by měl, i když v této fázi cyklu zatím nedochází ke konverzi, dbát na budování značky, jelikož už např. ve článku, který zákazníkovi pomůže s výběrem může odkázat na stránky se svými produkty nebo jemně zmínit svoji firmu. Zákazník poté bude mít nenásilně v povědomí kladnou zkušenost

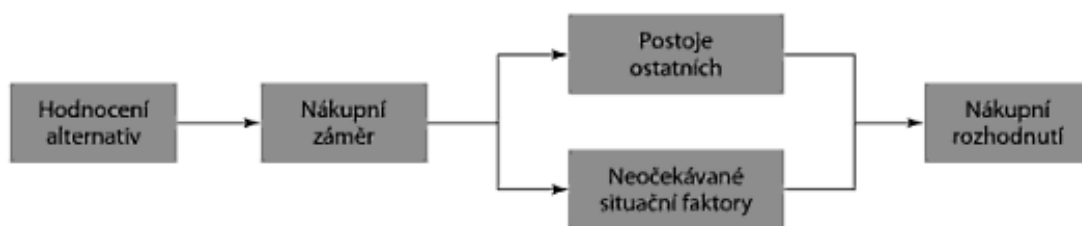
s touto značkou a může se při konečném rozhodování přiklonit k firmě, která mu nezištně pomohla tímto dobrým článkem (6).

### Hodnocení alternativ

Zákazník, který se dostal v nákupním cyklu až sem, již ví o své potřebě a vyhledal dostatek informací o daném produktu, ovšem v záplavě informací a značek či druhů výrobků má na výběr z nepřeberného množství možností koupě. Spotřebitel, který se snaží uspokojit svoji potřebu bude hledat výhody, jenž s nákupem daného výrobku získá. Pro zákazníka je výrobek seskupením vlastností, některé vlastnosti mají pro něj větší či menší význam, podle toho jak uspokojuje danou potřebu, jedná se o stupeň důležitosti vlastnosti. Jako marketéři bychom se měli věnovat vlastnostem důležitým pro zákazníka, zdůraznit je a tak se odlišit. Oproti tomu charakteristické vlastnosti jsou ty, které se zákazníkovi vybaví, když se mluví o produktu, nemusí být ale pro zákazníka důležité. Zákazník si vytváří hodnocení značky na základě svých dosavadních zkušeností. Také každá vlastnost vzbuzuje v kupujícím užitek, která může mít vliv na konečné rozhodnutí a zákazník si pomyslně sčítá celkový užitek, který mu výrobek přinese. Svoje stanoviska si utváří zákazník v celém proběhu hodnocení (5, str. 339-341).

Ve vyhledávání by tato část cyklu mohla vypadat tak, že uživatel zadává např. vyhledávací dotazy: „přednosti aut Audi“, „nejlepší auta roku 2012“, „recenze pevných disků“. Také je dobré, pokud se již zákazník dostane do našeho elektronického obchodu nabídnout mu porovnávání zboží s ohledem na vlastnosti, cenu, způsob určení.

### Nákupní rozhodnutí



Obrázek 3: Od hodnocení k rozhodnutí  
Zdroj: (5, str. 343)

Za normálních okolností povede nákupní rozhodnutí k nákupu oblíbené značky zákazníka. Někdy ale mezi nákupní záměr a nákupní rozhodnutí můžou vstoupit dva

faktory. Postoje ostatních osob, které mohou mluvit do rozhodnutí o koupi / nekoupi produktu, případně známých, kteří poskytnou věcnou radu a zvrátí rozhodnutí. Za druhé to jsou neočekávané situační faktory, pokud totiž zákazník počítá s jistými finančními prostředky pro měsíc nákupu nebo danou cenou produktu a tento faktor se v měsíci nákupu změní v jeho neprospěch, nebude si potom moci dovolit daný produkt pořídit a bude muset preferovat např. levnější variantu. Pokud nakupující pocítí úzkost nebo riziko ve svém rozhodnutí, může nákupní rozhodnutí zamítnout a vůbec samotný nákup neuskutečnit (5, str. 343).

Nákupní rozhodnutí v internetovém obchodě je většinou realizováno odesláním nákupního košíku a jeho objednáním či poptáním konkrétních služeb firmy pomocí webového formuláře.

### **Ponákupní chování**

Cyklus by mohl jednoduše minulým krokem skončit. Ovšem v dnešní společnosti, kdy se informace šíří rychlostí světla to není tak jednoduché – na internetu jsou většinou v tom samém obchodu, kde byl nákup proveden nebo na různých zbožíových srovnávacích možnosti uživatelských recenzí po provedeném nákupu. Když tedy prodejce přechválí svůj produkt a zákazník bude po nákupu zklamaný, jelikož nebyly naplněny jeho potřeby či očekávání, povede to k negativní reklamě. Na druhou stranu, když jsou očekávání překonána, může to vést k pozitivním ohlasům a v nejlepším případě k opakovanému nákupu (5, str. 344).

Je tedy třeba dbát na dobré jméno firmy a nesnažit se zákazníkům prodat za každou cenu nekvalitní výrobek nebo službu. V dnešní éře dostupnosti informací si totiž zákazník před každou transakcí najde dostupné informace o prodejci od něhož plánuje sjednat služby či nakoupit.

## 2.2 SEO - optimalizace pro vyhledavače

V poslední době každý člověk, který se vyskytuje na internetu, zná pojem internetový vyhledavač. Uživatel jeho pomocí vyhledává služby, produkty či kontakty na prodejce. Každá takto nalezená stránka, která je nám strojově podstrčena ve výsledcích, může mít za sebou optimalizaci, tj. že cílem bylo stránku zviditelnit a zvýšit její nalezitelnost. Otázkou je, zda-li když na výsledek hledání klikneme, opravdu nalezneme to, co jsme chtěli vyhledat nebo se dopravujeme k požadovaným informacím přes navigační strukturu webu a budeme z hlediska našich cílů jako uživatelé spokojeni. V následujících kapitolách si odpovíme na otázku, co je SEO, jak optimalizovat a navrhnout naše stránky s důrazem na uživatelskou přívětivost, cíle optimalizace, metody zvýšení návštěvnosti a měření výsledků.

### 2.2.1 Co je SEO?

Někteří lidé zaměřují optimalizace svých stránek pouze na první pozice ve výsledcích vyhledávání. Utratí za taková řešení nemalé prostředky a výsledek jim to nepřinese žádný – soustředili se jen na jednu část problematiky. Tito optimalizátoři užili taktické triky, řekněme dočasné znalosti děr v algoritmech vyhledávačů a zapomněli investovat svůj čas do dlouhodobějších řešení, která by pro ně pracovala několik let (7).

*„Kvalitní stránky, výborný obsah. Přesně to chtějí nalézat uživatelé a přesně to jim proto musí vyhledavače nabídnout. Ano, každý vyhledavač je jiný a jejich hodnotící algoritmy se neustále vyvíjejí. Všechny však směřují k jedinému cíli: nalézt a na prvních místech zobrazit stránky, které nejlépe uspokojí hledajícího uživatele.“ (8)*

Slovy Marka Prokopa, která použil ve svém článku z roku 2003 by se dalo popsat i SEO v roce 2013. Snad jen podle jeho rozhovoru (9) by se termín „*optimalizace pro vyhledávače*“ změnil na „*optimalizace nalezitelnosti na internetu*“. Podle stejného článku by se také dalo říci, že „*SEO je optimalizace pro uživatele*“. Cílem našich stránek je spokojený uživatel a vyhledavače se snaží čím dál více přemýšlet jako lidé, uděláte-li vynikající stránky s dobrým obsahem, vyhledavače to vezmou v potaz.

V kontextu celého webu do SEO patří disciplíny, které se zaměřují na faktory nacházející se na samostatných stránkách – klíčová slova, texty, zdrojový kód stránek či informační architektura či grafický design. Faktory mimo web, tzv. offpage jsou zejména zpětné odkazy a bude jim věnována také podkapitola. Všechny tyto techniky je třeba užívat v zájemné synergii.

### **2.2.2 Jak uživatelé vyhledávají**

Než přistoupíme k problematice optimalizace, měli bychom nejprve zjistit, jak na internetu lidé vyhledávají. Každý uživatel zadává do fulltextu nějakou frázi a očekává, že mu budou výsledky interpretovány v co nejvyšší míře relevance k jeho dotazu, tzn. dostane to, co požaduje. Vyhledavač zpracuje dotaz a vrátí na výstup výsledky, které jsou řazeny podle určitých priorit, také se snaží k dotazu přiřadit nějaký úmysl a dále s ním tak pracuje. Dotazy si můžeme klasifikovat následovně (10, str. 74):

- Navigační – hledající se snaží najít konkrétní stránku, většinou se jedná o značky produktů nebo firmy, lze ale narazit na vyjimku. Příklad dotazu: „mercedes“, „nutella“, „aukro.cz“.
- Komerční (transakční) – nejčastěji takto zadaný dotaz souvisí s prodejem nějakého výrobku nebo služby, algoritmus tedy upřednostní výsledky, kde lze provést koupi nebo poptávku. Příklad dotazu: „povinné ručení on-line“, „nákup bundy outdoor“, „horské kolo e-shop“.
- Informační (průzkumný) – smyslem tohoty typu dotazu je najít nějakou konkrétní informaci, která nemusí mít ani komerční charakter. Vyhledavač upřednostní výsledek, který bude mít informační smysl a bude nejlépe odpovídat na položený dotaz, nebude upřednostňovat prodejní dotazy. Příklad takového dotazu může být: „datum narození Jana Ámose Komenského“, „druhá světová válka“
- Průzkum před nákupem – je to jakási podmnožina informačních vyhledávacích dotazů, rozdíl je zde ale takový, že skupina lidí využívající tento typ, zvažuje koupi výrobku či služby a nachází se v dřívější než konverzní části nákupního

cyklu. Pokud oslovíme vyhledávající v této fázi, máme větší šanci, že nákupní rozhodnutí bude provedeno na našich stránkách. Příklad vyhledávání: „recenze ledniček 2013“, „srovnání moderních hraček pro děti do 5 let“.

Vyhledavače se snaží rozpoznat typ vyhledávacího dotazu, který uživatel zadal – může k tomu používat svoje historická data, kde bere v potaz statisticky významnou veličinu – miliony vyhledávání v minulosti, také může hodnotit dotaz podle uživatelského chování v minulosti. Velké procento okamžitých odchodů ze stránek u některých výsledků napovídá vyhledavači, že se asi nebude jednat o kvalitní obsah.

### Metoda dlouhého ocasu – longtail

Ve skutečnosti menší procento dotazů, které zadáváme jsou jednoslovné (23%) nebo dvouslovné (23%), většina dotazů, které jsou napsány do vyhledávače jsou tři a víceslovné (tříslovné – 20%, čtyřslovné – 14%, pětislovné – 8.6%, atd.) (11).

Technika co nejširšího pokrytí uživatelských dotazů a frází se nazývá „metoda dlouhého chvostu“.

*„Jedná se o přesunu zájmu z podstatě malého počtu hitů k většímu množství okrajových předmětů (1, str. 30)“*

Ve vyhledávání to bude např. přesun zájmu od obecného dotazu: „vysavač“ ke konkrétnímu: „vysavač Electrolux ZJM 6820 Jetmaxx“.



Obrázek 4: Strategie dlouhého chvostu  
Zdroj: (12)

Tato metoda má nejvyšší využití hlavně u internetových obchodů, kde je ocas hodně dlouhý a jeho pokrytí má větší návratnost než obecné dotazy, jelikož při zadání konkrétního dotazu máme větší šanci ke konečné konverzi, nehledě na to, že obecný dotaz může být položen za účelem zodpovězení informace a ne s nákupním motivem (např. „postel recenze“).

### **2.2.3 Jaké jsou cíle SEO**

Cíle optimalizace by měly být totožné s cíly našich internetových stránek / obchodu (7). Je rozšířeným mýtem, že cílem by měla být pouze větší návštěvnost z vyhledávačů. Laikové považují za cíl první pozici na jedno slovo a nezajímají se o širší kontext. Pravdou ovšem je, že návštěvnost je jenom prostředek, nikoliv cíl. Cílem je zviditelnit web tak, aby na něj přicházelo co nejvíce zacílených návštěvníků ovšem - cíle webu jsou poměrně často definovány peněžně, ale nemusí to být pravidlem.

Cíle SEO jsou většinou:

- zvýšení zisku (13)
- prodej zboží, služeb
- získání poptávky
- nábor nových zaměstnanců
- budování značky
- získání peněz z kontextové reklamy

### **2.2.4 On-page faktory SEO při realizaci webu**

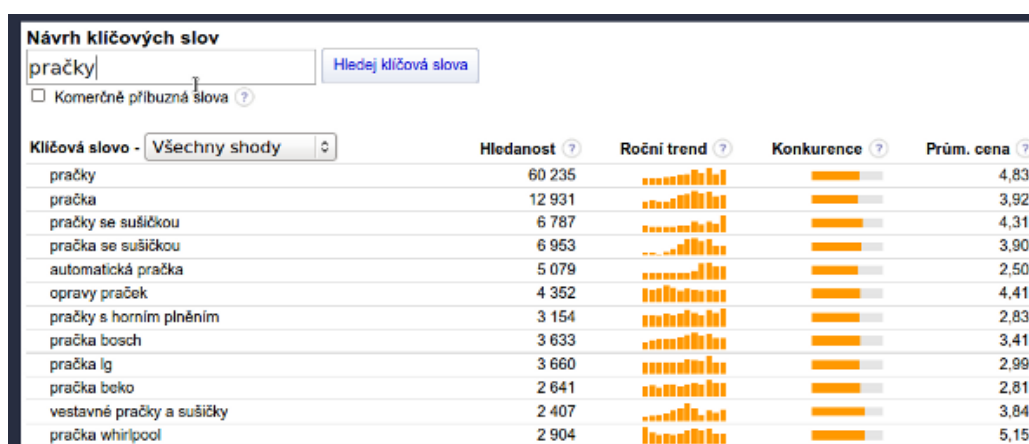
Níže si probereme faktory, které mohou ovlivnit návštěvnost z vyhledávačů, potažmo funkčnost celého webu. U kapitoly klíčových slov se zaměříme na to, jak uspokojit poptávku a nevnučovat uživatelům to, co nechtějí. Poté si projdeme prvky, které by měl obsahovat správně naprogramovaný web, bez problémů indexovaný vyhledávačem. Následně se budeme věnovat problematice, jak správně rozmístit klíčová slova (obsah) uvnitř webového dokumentu. Informační architektura hraje pro uživatele nemalou roli a ani v SEO na ni nemůžeme zapomenout, proto ji věnujeme menší kapitolu.

## Klíčová slova na stránkách

„Klíčové slovo je textový řetězec tvořený jedním nebo více slovy.“ (14)

V předcházející části „Jak lidé vyhledávají“ jsme si pověděli o druzích vyhledávacích dotazů. Při analýze klíčových slov bychom měli dbát na to, v jaké fázi nákupního cyklu se uživatel nachází. Měli bychom těmto druhům dotazů přizpůsobit náš obsah, ať se již jedná o dotazy komerční, informační či průzkumné, vždy bychom měli uživateli nabídnout adekvátní cílovou stránku, pokud máme pouze vysokou pozici na naše klíčové slovo i odpovídající míru prokliku z vyhledavače a uživatel na našich stránkách nenajde to, co požaduje, stránku neprodleně opouští.

## Návrh klíčových slov



Obrázek 5: Návrh klíčových slov v systému Sklik.cz

Zdroj: (15)

Jako marketéři, kteří vymýšlí celou strategii webové prezentace, jsme obeznámeni s produkty či službami, jenž jsou cílem zájmu našeho prodeje a propagace. Nástroj pro návrh klíčových slov v PPC systému Sklik.cz by byl snem reklamních agentů minulého století. Jeho pomocí se dá bez problému najít počet vyhledávacích frází, které zadávají uživatelé do českého fulltextového vyhledavače *Seznam.cz*. Můžeme tak uživatelům nabídnout opravdu to, co je pro ně relevantní. Výhodou je lokalizace v českém jazyce a množství komerčně příbuzných frází, které nám po zadání vygeneruje s ohledem na frekvenci hledání. Můžeme tak svoje stránky pokrýt více konkrétními a tudíž méně konkurenčními frázemi.



## Analýza klíčových slov

V předchozím kroku jsme vybrali slova, která jsou pro náš obor relevantní. Každá skupina vyhledávacích dotazů by měla mít svoji vstupní stránku, někdy může stránka odpovídat vstupnímu dotazu. Měli bychom také položit důraz na synonyma daného klíčového slova a vertikální (slova, která spolu úzce souvisejí) a laterální analýzu (slova spolu zdánlivě nesouvisejí, ale vytvářejí dohromady jedno téma).

Analýzou konkurence sice zjistíme, na která klíčová slova se oponenti umisťují, ale nemáme žádnou možnost zjistit, zda-li se jim takovéto pozice finančně či jinak vyplácí (7). Může nám to posloužit, pokud jsme na nějakou skupinu slov zapoměli. Také se budeme muset snažit najít vzorce skládání dalších méně běžných dotazů. Z analýzy klíčových slov poté máme zpracovatelný výstup pro návrh informační architektury. Kolik klíčových slov se vejde na jednu stránku se můžeme také dočíst v článku Marka Prokopa (16).

## Kam umístit klíčová slova

SERP – Search Engine Result Pages, je to stránka, kde jsou zobrazeny výsledky vyhledávání pomocí fulltextových vyhledávačů *Seznam.cz*, *Google.cz* (17, str. 24).



**Lednička aneb lednice ELECTROLUX, levneELEKTRO.cz**  
Lednička kombinovaná s mrazničkou dole **ELECTROLUX** EN3241AOW s objemem chladničky 223 litrů a mrazničky 78 litrů. Zaujme také energetickou třídou ...  
[www.levneelektro.cz/s33-lednicka?vr=ELECTROLUX](http://www.levneelektro.cz/s33-lednicka?vr=ELECTROLUX)

Obrázek 6: Výsledek v SERP Seznam.cz  
Zdroj: (18)

## Lednice Electrolux - Heureka.cz

[electrolux.heureka.cz](http://electrolux.heureka.cz) › Značky › Electrolux

**Lednice značky Electrolux.** Kompletní informace o produktech, nejnižší ceny z internetových obchodů, hodnocení, recenze.

Obrázek 7: Stránka v SERP Google.cz  
Zdroj: (19)

Když se podíváme na první výsledek vyhledávání („ledničky electrolux“) viz Obrázek 6 z vyhledávače *Seznam.cz*, resp. Obrázek 7 z *Google.cz*, vidíme, že v titulku výsledku je uveden název zboží a brandový název obchodu. Poté je uveden jemný výstřížek textu,

potažmo url domény a snippet drobečkové navigace v druhém případě a až poté vlastní text. Kdybychom se podívali do zdrojového kódu cílové stránky, zjistili bychom, že se tyto části vyhledávání nacházejí v uzavřené v html elementech, a adresa je viditelná jasně v url. Tímto jednoduchým zjištěním můžeme říci, že vyhledavače také kladou váhu na ty části webové stránky, které jsou viditelné v SERP. Není to samozřejmě jediné měřítko, ale pro rozmístění klíčových slov použijeme části html dokumentu zmíněné níže. Jedná z částí o sémantiku webu – měli bychom dodržovat pravidla přístupnosti a validity stránek a dodržovat webové standardy.

**Titulek stránky** – titulek je na každé stránce originální.

*„Když bychom měli optimalizovat pouze jedinou část stránek, měl by to být titulek.“ (7)*

Do titulku je dobré umisťovat důležitá klíčová slova, následovaná názvem firmy za oddělovačem, jelikož takovýto výsledek vypadá v SERP více důvěryhodně (20, str. 121-123).

**Nadpisy (h1-h6)** – nadpisy nám strukturují jednotlivou stránku do více částí, umožňují zvýraznit myšlenky obsahu a oddělit jednotlivé kapitoly. Bezesporu asi největší váhu má nadpis první úrovně - h1. Je dobré zahrnout klíčová slova do nadpisů napříč celou stránkou, nadpisy by měly vyznívat přirozeně a neměly by se zaspamovávat (tapetovat) klíčovými slovy.

**Meta description** – úryvek textu, který si občas při nějakém dotazu vybere vyhledavač google do výsledku vyhledávání, nám může zvýšit proklikovost výsledku, fulltext Seznamu na něj moc nedbá. Pokud optimalizujeme pro Google, tak bychom tomuto tagu a jeho obsahu měli věnovat zvýšenou pozornost.

**Url adresa** – rozmístění klíčových slov v doméně, potažmo v adrese má určitě svoji váhu. Více slov může být odděleno pomlčkou pro snazší čitelnost, ovšem v dnešní době vyhledavač rozpozná i spojená slova bez pomlčky. Klíčové slovo v doméně má takovou váhu do té míry, dokud je zapamatovatelné.

*„Představte si kdyby najednou vypnuli všechny vyhledavače, kam lidé budou chodit nakupovat?“ (7)*

Jelikož se vyhledavače snaží přizpůsobovat svoje výsledky lidem, klíčové slovo v názvu domény např. „lednický.cz“ by pro ně mělo lepší zvuk než „levné-bílé-lednický.cz“. Čím víceslovný název je, tím menší přikládá váhu jednotlivým slovům. Ostatně jako u všech předchozích tagů. Přesmyčkou těchto slov by vzniklo nepočítaně kombinací, tudíž i zde platí „méně znamená více“. Klíčová slova po názvu domény za lomítkem už nemají takovou váhu jako v doméně, ovšem najdeme pro ně také využití. Měli bychom se vyvarovat pokusů vkládat navigační strukturu do url (7). Špatný příklad takových pokusů je: „*pracky.cz/pracky/predni-plneni/ecg-ewf-851-m*“ místo toho je třeba použít: „*pracky.cz/ecg-ewf-851-m*“. Takovýto přístup nám zbytečně řadí sílu jednotlivých slov, optimalizujeme přeci na konkrétní typ pračky: „*ecg-ewf-851-m*“ (metodika long tail) a ne každou stránku webu na „*pračky přední plnění ecg-ewf-851-m*“.

**Obsah dokumentu** – klíčové slovo, pro které optimalizujeme, by se mělo jednou či dvakrát určitě vyskytovat v dokumentu, pokud je hodně důležité tak klidně i vícekrát. Nemusíme si měřit hustotu klíčových slov v procentech a používat jiné „užitečné“ šamanské recepty. Měli bychom dbát na pestrost jazyka napříč celou stránkou a nepoužívat pouze první pády. Jelikož vyhledavače mají vlastní slovníky synonym, provádějí si laterální a vertikální indexování, je lepší když každá jednotlivá stránka pokryje celé jedno či část tématu s použitím všech dostupných jazykových možností. Samozřejmě má váhu blízkost slov a jejich pořadí.

**Menu, navigace** – jednoduše řečeno klíčová slova slova v těchto prvcích pomáhají uživateli dát si do kontextu to, na kterou stránku se dostane, pokud je použije. Zároveň plní funkci interních zpětných odkazů.

**Popisek obrázku** – v rámci použitelnosti by každý obrázek měl obsahovat popisek. Vyhledavač si neumí obrázek zobrazit a tak je popisek jediná možnost jak dát vědět uživateli co obsahuje.

**Text zpětných odkazů** – v textu interních i externích zpětných odkazů (položka menu je také zpětný odkaz) (17, str. 34-40), (20, str. 119-131), (21, str. 48-51).

## **Programování webu**

Při vytváření internetových stránek musíme vzít v potaz několik hledisek - naprogramování webu je sice technická záležitost, ovšem návrh informační architektury může být záležitostí UX návrháře či SEO konzultanta. V další podkapitolách si popíšeme tuto problematiku.

### **Informační architektura**

*„Úkolem informační architektury je snadná dohledatelnost produktů, služeb a informací na našich webových stránkách.“ (22)*

Návrh informační architektury z pohledu SEO není triviální záležitostí a je ji třeba věnovat zvýšenou pozornost. Pokud máme dobře zpracovanou analýzu klíčových slov, může její výsledek posloužit pro návrh IA.

Klíčová slova rozdělíme do skupin, podle kterých v následujícím kroku vytváříme strukturu navigace našeho webu. Použitím různých typů navigace můžeme kompenzovat konkurenčnost klíčových slov, jelikož každý prvek navigace je také zároveň zpětným odkazem, zvýšením počtu interních odkazů na naše podstránky vylepšíme pozici ve vyhledávání. Samozřejmě bychom to neměli s počtem odkazů přehánět, ale nalezení optima závisí na potřebách a zkušenostech návrháře.

Informační architektura by měla umožňovat zaindexovat vyhledávačem všechny stránky na našem webu a také by měla poskytnout tzv. vstupní (landing) stránky pro každé klíčové slovo či soubor frází, přes které chceme být nalezitelní. Měli bychom se také vyvarovat duplicitního obsahu (stejně stránky dostupné pod jinou url) (23).

Rozmístění klíčových slov v prvcích na stránce jsme se věnovali v minulé kapitole – jsou také součástí IA. Z pohledu struktury webu bychom měli zvláštní důraz klást na navigaci našich webových stránek. Jedná se o jakési logické rozdělení celé prezentace.

6 základních druhů navigace na webu (24):

- hierarchická
- globální

- lokální (kontextová)
- odkazy v textu
- mapa webu
- interní vyhledávání

Před vlastní realizací webu bychom se měli u každého typu navigace zamyslet a zjistit, které řešení je pro nás nejvhodnější, a zda daný typ navigace do projektu začlenit. Je třeba připomenout základní pravidlo tvorby uživatelsky přívětivých stránek: návštěvník by měl vědět v každém okamžiku, kde se v dané struktuře webu nachází (25). Pro rozřídění rozsáhlejších projektů a filtrování stránek se dá také použít facetová klasifikace, podrobný návod, jak ji navrhnout, můžeme nalézt v článku (26), který se jí důkladně zabývá.

### **Technické překážky indexace**

To, aby robot zaindexoval - jednoduše načetl naše stránky do svých databází a poté je nabízel ve výsledcích vyhledávání je alfa optimalizace pro vyhledavače. Pokud stránky nejsou nalezitelné, nedá se posléze hovořit o zvyšování pozic, jelikož jednoduše není co zvyšovat. Mezi základní prohřešky při tvorbě stránek patří (27):

- ryze javascriptové, flashové stránky
- používání iframes
- duplicitní obsah (zmíněný v předchozí kapitole)
- velká velikost kódu
- neošetřené neexistující stránky (error 404), špatně nastavené přesměrování
- neexistuje sitemapa webu

Některé neduhy stránek se dají hlídat pomocí různých validátorů či výstupních testů stránek. Při rozsáhlejších stránkách se doporučuje použití Google Webmaster Tools, které jsou schopny upozornit správce webu na některé technické nedostatky. Chybová stránka 404 by se měla zobrazit pod unikátním url – nemít jednu adresu, kam se přesměrovávají všechny chyby (7).

## **2.2.5 Offpage faktory – linkbuilding**

Ve chvíli, kdy máme odladěny všechny přecházející kroky optimalizace a pokud jsme v konkurenčním segmentu, máme zatím hotovo asi 5% veškeré práce (7). Přichází na řadu metoda zvaná linkbuilding.

Pojem linkbuilding by se dal česky přeložit a popsat jako budování zpětných odkazů na naše webové stránky. Přes tyto odkazy k nám proudí návštěvnost a nepřímo tyto odkazy zvyšují pozice stránek ve vyhledávačích. Nejde o to získávat odkazy z jakéhokoli typu webu, ale je třeba vytvářet kvalitní odkazové portfolio, které může diverzifikovat změny vyhledávacích algoritmů. Měli bychom se starat o tématickou příbuznost – ze zpětného odkazu by měl mít uživatel nějaký užitek, když budeme mít webové stránky o autech a budou k nám směřovat odkazy z článků o ledničkách, těžko to bude relevantní. Ideální zpětný odkaz je např. popsán na firemním blogu Medio Interactive (28) – vzpomenu nejdůležitější vlastnosti – odkaz je z autoritativní stránky, tématicky příbuzný, dlouhodobý, viditelný text se shoduje s našimi klíčovými slovy a frázemi a odkaz není koupený.

### **Postup budování odkazů**

Podle Janoucha (1, str. 126-127) můžeme rozdělit proces budování zpětných odkazů do několika kroků:

- zjištění, kdo na nás odkazuje
- analýza zpětných odkazů konkurence
- zjištění webů, ze kterých chceme získat zpětný odkaz
- vlastní získání odkazů
- monitorování výsledků

### **Kdo na nás odkazuje a zpětné odkazy konkurence**

Pokud vytváříme web, měli bychom se přesvědčit o naší nově koupené doméně, zda-li není nějakým způsobem vyřazená z výsledků vyhledávání – zabanovaná, v případě menších projektů a běžných domén je pravděpodobnost malá, ovšem pokud doménu

kupujeme v aukci či za nemalé peníze, je třeba tuto hrozbu ověřit. Také v případě převzetí zakázky od jiné marketingové firmy je třeba zhodnotit odkazový profil.

Konkurence je z hlediska zpětných odkazů, pokud se umisťuje na lukrativní klíčová slova, případně má pokrytý dlouhý chvost, je na tom lépe než my v okamžiku startu projektu. Měli bychom si provést analýzu zpětných odkazů konkurence pomocí dostupných nástrojů předtím, než si určíme vlastní linkbuildingovou strategii. Můžeme použít placené dostupné nástroje nejen pro analýzu odkazů: *Majestic SEO*, *Raven Tools* či české *Collabim*.

### **Odkud získat zpětný odkaz**

Poté, co jsme si provedli analýzu zpětných odkazů konkurence, musíme si vyhodnotit odkud budeme získávat zpětné odkazy. Slepé následování strategie našich rivalů se nemusí vždy osvědčit, nevíme jestli jim zvolená metoda přináší nějaký užitek, jelikož nemáme srovnávací údaje výnosů a nákladů. Měli bychom se tedy spíše zaměřit na budování vlastních metod a pokusit se vymyslet něco nového, originálního, snažit se poskytnout kvalitní služby a provázat tak interní firemní vystupování s online vztahy. Zde bych chtěl pouze upřesnit, že již několik let nemá cenu tolik sledovat Pagerank stránek, kterýžto ukazatel nám spíše udává jakousi popularitu – sic hrubý počet odkazů vedoucích na stránku, ale nemůžeme na něm stavět svoje strategie a doporučoval bych použití spíše orientační.

Ideální stránky, ze kterých chceme získat odkaz se většinou mohou nacházet v pořadí nad našimi stránkami, na určité vyhledávací fráze. Diverzifikací zdrojů zpětných odkazů získáme kvalitní odkazový profil, o kterém se můžeme také dočíst více na blogu agentury MEDIO Interactive (28).

### **Metody získání zpětných odkazů**

Pokud máme ryze komerční stránky, je těžké pro ně získávat přímé zpětné odkazy z toho důvodu, že se na nich většinou nachází popis naší firmy a služeb – proč by na nás měl někdo odkazovat. Je tedy jakousi podmínkou mít odkazovatelný obsah, toho se dá docílit např. psaním profesního blogu či různých zajímavých návodů a poté si z nich odkázat na svojí prezentaci. Odkazovatelný obsah může tedy např. být:

- informace – články, zpravodajství, návody
- fotografie
- aplikace, mobilní hry
- blog
- linkbaitingová kampaň
- sociální profil (Facebook, Google+, Twitter)

Na všechny uvedené techniky můžeme zapomenout a začít odkazy klidně nakupovat či obesílat tisíce webmasterů a žádat je o umístění našeho odkazu na jejich stránkách, ovšem takovouto taktikou budeme pouze ztrácet čas. Jelikož časem určitě vyhledavače odhalí nakoupené zpětné odkazy a odpovědi na zprávy nám přijde mnohem méně, než jsme čekali a výsledky budou tristní.



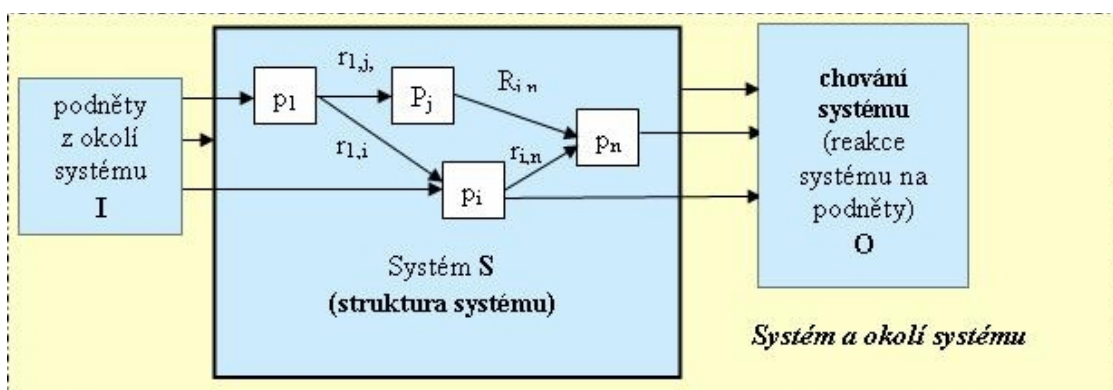
## 2.3 Elektronický obchod

V této kapitole se budeme věnovat elektronickému obchodování a jeho formám. V úvodu si vymezíme systémově model obchodu, v následující pasáži se zaměříme na druhy modelů elektronického obchodování.

### 2.3.1 Systém a model elektronického obchodu

#### Systém

Během studií různých jevů a procesů – objektů bylo vědci zjištěno, že různé procesy a postupy použité v jedné oblasti se dají s úspěchem při určitém zjednodušení použít i při zkoumání oblastí jiných. Pokud tedy zobecníme v těchto pozorováních vztahy mezi objekty, chování a jejich vzájemnou interakci, vznikne nám nová disciplína – obecná teorie systémů. Terminologie, vypracovaná pro teorii systémů nám umožňuje zkoumání různých objektů při použití stejného jazyka. Systém můžeme tedy považovat za abstrakci těchto objektů. Vlastním jádrem teorie systémů bude tedy soubor abstraktních objektů (29, str. 4-5), viz Obrázek 8.



Obrázek 8: Systém a okolí systému

Zdroj: (29, str. 6)

„Systém  $S = \{P, R\}$

*je účelově definovaná množina prvků  $P$  :*

$P = \{p_i\}$ , kde  $i \in J$  ( $J$  je množina indexů)

*a množina vazeb (vztahů)  $R$  :*

$$R = \{r_{i,j}\}, i, j \in J \text{ mezi prvky } p_i \text{ a } p_j \text{“ (29, str. 6)}$$

*„Prvky  $p_i$  systému  $S$  jsou jeho elementární části. Množinu  $P$  všech prvků  $p_i$  nazýváme universum systému. Vazby jsou vzájemné závislosti mezi prvky  $p_i$  a  $p_j$  nebo vzájemné působení mezi těmito prvky. Může jít o informační vazby, vyjádřené vztahy.“ (29, str. 6)*

Množství jednotlivých prvků v systému můžeme nazvat složitost. Čím více si identifikujeme vazby uvnitř systému, tím jdeme více do hloubky a objevujeme vzájemné vazby. Hloubkou poznání rozumíme rozlišovací úroveň. Prvky, které nepatří přímo do systému, ale mají na něj vliv, pojmenujeme okolí systému. Z vnějšku působí na systém podněty (vstupy) a systém naopak působí na okolí pomocí odezev (výstupů). Systém můžeme vytvořit z více podsystémů. Systémy si můžeme různě škálovat, ovšem nás budou zajímat hlavně systémy ekonomické a informační (29, str. 7).

Systém si můžeme popsat podle Dvořáka (29, str. 7-8):

- strukturou systému
- chováním systému
- strukturou i chováním systému

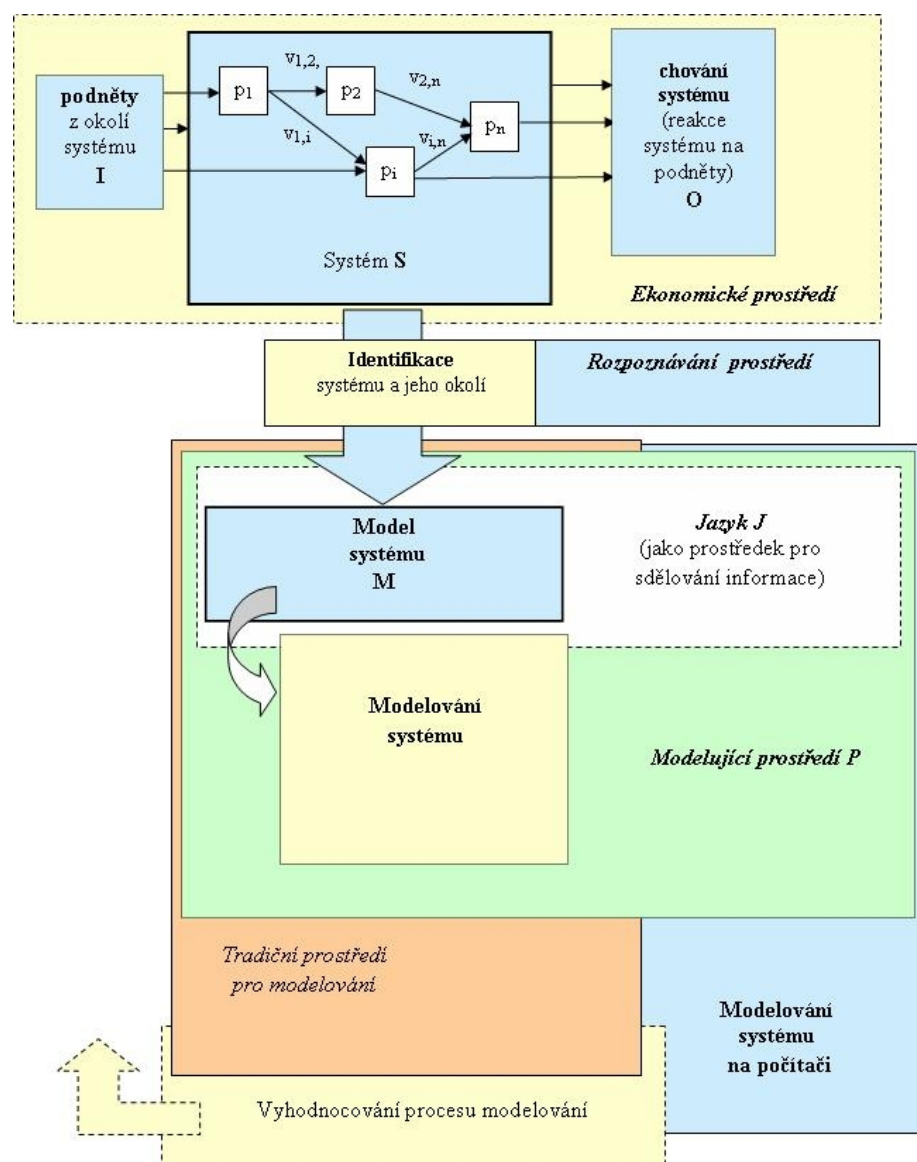
*„Získané údaje musíme obvykle někomu sdělovat. Prostředkem pro sdělování informací je jazyk. Mateřský jazyk se vyznačuje příliš velkou nejednoznačností - proto slovní (verbální popisy) systémů jsou velmi komplikované. Jsou však zatím nejužívanější zejména při popisu tzv. sociálních a dalších systémů.“ (29, str. 8)*

*„Podstatně kvalitnější jsou jazyky umělé, jednoznačné a těmi mohou být jazyky využívající prostředků matematiky. Takže proces popisu systému  $S$  vede přes jeho definování k vytváření jeho vlastního modelu  $M$ .“ (29, str. 8)*

$$S \rightarrow M$$

*„V procesu identifikace (zkoumání) systému jde o vytváření modelů různých tříd. Tomuto procesu říkáme modelování. Pojem model lze chápat jako možnou realizaci vhodné teorie nebo jako jisté zobrazení systému. Pro další modelování může být model také chápán jako jistý systém.“ (29, str. 8)*

## Model



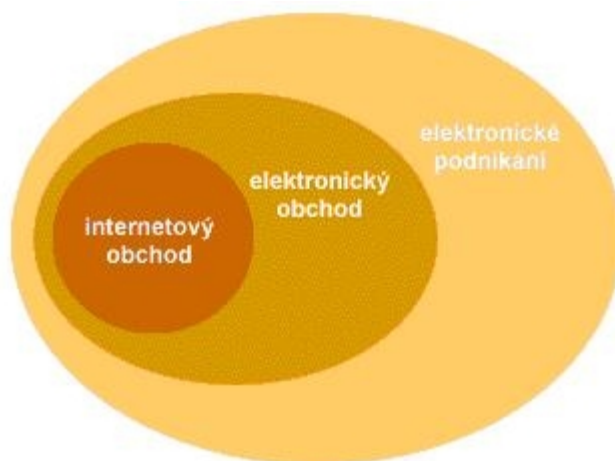
Obrázek 9: Modelování systému  
Zdroj: (29, str. 15)

Na zvolené rozlišovací úrovni je možné realizovat ekonomický model  $M$  a tvorba tohoto ekonomického systému bude vázaná jazykem  $J$  (v našem případě ekonomické názvosloví), případně jazykem umělým, viz. Obrázek 9. Cílem identifikování je rozlišit a popsat zvolený systém. Jelikož tyto modely jsou dynamické, jsou vždy identifikovány v čase  $t_k$ , v okamžiku zkoumání. Modelující prostředí  $P$  nám vždy omezuje svými podmínkami možnosti a rozsah naší simulaci ekonomického prostředí. Na obrázku 8 můžeme nalézt tradiční prostředí pro modelování a také poslední dobou používané

modelování v počítačovém prostředí. Každý takovýto model má také zpětnou vazbu do existujícího prostředí ekonomického systému. Vše musí probíhat v reálném čase. Výsledky těchto procesů používáme k upřesňování našeho modelu. Identifikace systému probíhá v těchto krocích – rozpoznání objektu (a jeho situace), jeho zjednodušení a definování (rozpoznání) systému (29, *str. 12-15*).

### 2.3.2 Druhy elektronického obchodování

*„Elektronické podnikání dnes představuje celou škálu produktů, aplikací a služeb, jejichž společným jmenovatelem je využití elektronických komunikačních kanálů a zejména infrastruktury Internetu pro realizaci obchodních procedur a operací.“*  
(29, str 20)



Obrázek 10: Elektronické obchodování  
Zdroj: (29, str. 21 )

Elektronický obchod je podmnožinou elektronického podnikání. Elektronické podnikání, viz Obrázek 10, zahrnuje všechny možné digitální operace při realizaci komerčních transakcí, rozumí se tím využití všech technologií pro vylepšení firemních procesů (29, str. 8). Oproti tomu je elektronický obchod jakoukoli formou obchodování, která je prováděna elektronicky – použití elektronického média s účelem uzavřít obchodní transakci (29, str. 9). Podmnožinou elektronického obchodu je internetový obchod.

Podle Švadlenky (30, str. 17) máme 4 základní obchodní modely elektronického obchodování:

- B2B (Business to Business)
- B2C (Business to Customer)
- C2B (Customer to Business)
- C2C (Customer to Customer)

V současné době ale vznikají další obchodní modely (30, str. 17):

- A2B (Administration to Business) - G2B (Government to Business)
- A2C (Administration to Citizen ) - G2B (Government to Citizen)
- B2A (Business to Administration ) - B2G (Business to Government)
- C2A (Citizen to Administration ) - C2G (Citizen to Government )

### **Elektronický obchod B2B**

Tento model je postaven na vzájemném obchodování mezi firmami. V tomto typu prodávající svého odběratele většinou velice dobře zná. Je kladen důraz na předem vyjednané obchodní podmínky či snížení administrativní zátěže. Vztahy mezi partnery se dlouhodobě budují a jsou smluvně podchyceny. B2B můžeme dále rozdělit na 3 podtypy (30, str. 17-18):

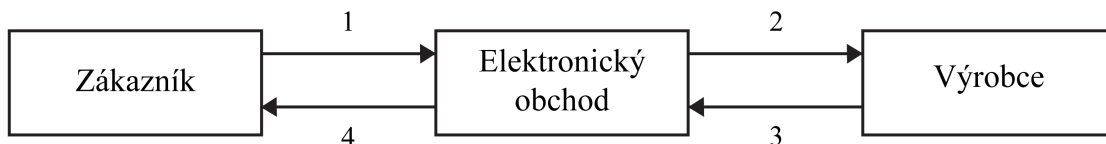
- seller centric – velkoobchod distribuuje svoje zboží více maloobchodům
- buyer centric – jinak také e-procurement (zásobování), více dodavatelů soutěží o jednu zakázku odběratele
- e-marketplace – jinak také elektronické tržiště, kde se setkávají odběratelé i dodavatelé, funguje podobně jako burza.

### **Elektronický obchod B2C**

*„Jde o obchodní vztahy mezi podnikem a konečným spotřebitelem, realizované webovými aplikacemi, virtuálními obchody na Internetu a podobně.“ (29, str. 23)*

Výhodou tohoto modelu je, že obchodník má o zákazníkovi uložené informace, které může použít pro marketingové účely, dále je také schopen díky zjednodušení procesu snížit cenu koncového produktu. K nesporným pozitivům patří také detailní popis a vyčerpávající informace o jeho službách a prodejních artiklech. Ideální fungování elektronického obchodu (Obrázek 11) si můžeme popsat v následujících krocích (30, str. 21-22):

1. objednávka zboží
2. kontaktování výrobce objednaného zboží
3. dodání zboží od výrobce do obchodu
4. expedice zboží koncovému zákazníkovi



Obrázek 11: Princip fungování elektronického obchodu

Zdroj: (30, str. 22)

Obchodování v B2C můžeme dále rozdělit na několik dalších druhů (29, str. 23):

- prodej informací
- prodej a pronájem
- prodej zboží
- poskytování reklamního prostoru

### **Elektronický obchod C2C**

V tomto modelu je typické, že ceny tvoří uživatelé systému, respektive určují maximální částku, kterou chtějí za zboží utratit. V modelech C2C může být prodávající zároveň i nakupujícím, obchody probíhají většinou principem aukce. (29, str. 24).

## 3 ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE

### 3.1 Běžná funkcionalita e-shopu

Internetové obchodování klade důraz na zákaznickovy potřeby a zájmy. Pokud máme kvalitní produkt, neznamená to, že se automaticky bude bezpracně prodávat. Když už návštěvník otevře naši webovou aplikaci, je třeba ho také zaujmout a zbytečně mu např. navigací, uspořádáním prvků či základními úkony neznepříjemňovat pohyb napříč našimi stránkami. V této části bych se chtěl zabývat základní funkcionalitou každého internetového obchodu a poté také některými rozšiřujícími možnostmi, které mohou sloužit k podpoře prodeje, provázání obchodu s jiným informačním systémem, apod.

#### 3.1.1 Frontend

Frontend internetové aplikace je jednoduše řečeno ta část, se kterou přichází běžný návštěvník do styku. Tato část se může skládat z celé řady funkcionalit, které si postupně popíšeme pro aplikaci internetového obchodu. Jedná se zejména o možnosti registrace zákazníka, nákupu zboží či pouhé prohlížení produktů.

#### Registrace

Nová registrace

Údaje o kontaktní osobě	Fakturační a hlavní dodací adresa
e-mail: <input style="border: 1px solid #ccc;" type="text" value="mail@davidstransky.cz"/>	ulice (včetně čísla): <input style="border: 1px solid #ccc;" type="text" value="Lonkova 465"/>
jméno: <input style="border: 1px solid #ccc;" type="text" value="David"/>	obec: <input style="border: 1px solid #ccc;" type="text" value="Pardubice"/>
příjmení: <input style="border: 1px solid #ccc;" type="text" value="Stránský"/>	PSČ: <input style="border: 1px solid #ccc;" type="text" value="53009"/>
telefon: <input style="border: 1px solid #ccc;" type="text" value="+420732498048"/>	stát: <input style="border: 1px solid #ccc;" type="text" value="Česká republika"/>
registrovat jako: <input checked="" type="radio"/> fyzická osoba <input type="radio"/> firma	

Údaje o účtu

přihlašovací jméno:

heslo:  [Středně silné heslo](#)

ověření hesla:

☒ souhlasím se zasíláním marketingových materiálů od společnosti CZC.cz s.r.o. na výše uvedený e-mail

(\*) Údaje označené hvězdičkou jsou povinné

[Zaregistrovat](#)

Obrázek 12: Příklad registrace v e-shopu CZC.cz

Zdroj: (31)

Pokud návštěvník pouze prohlíží zboží či obsah stránek a jiné informace, neměli bychom ho nutit se registrovat do našeho systému – zatím vystupuje jako



neregistrovaný. V případě, že projeví zájem o objednání produktu a jeho následné zaslání, měli bychom mu nabídnout jednoduchý registrační formulář, který by neměl uživatele nutit zadávat žádná inkriminující osobní data jako jsou rodná čísla, kódy kreditních karet, apod. Můžeme zvážit, zda-li v tomto kroku chtít fakturační údaje, adresu nebo pouze poskytnout přístup do našeho systému, po zadání uživatelského jména nebo e-mailu a hesla. Zadání adresy a ostatních údajů můžeme přesunout do až kroku objednávky.

## **Katalog**

Katalog v internetovém obchodě slouží k procházení výrobků, jejich řazení či jiným operacím např. přidání do košíku, přidání do knihy přání, hlídací pes a jiné doplňkové funkce.

## **Kategorie**

Jako základní prvek navigace po produktovém katalogu slouží kategorizace zboží. Pokud by byly všechny výrobky v jednom výpisu, těžko by se v nich zákazník orientoval. Takto může majitel webu velmi jasně určit kam který výrobek patří a zákazník postupně prochází kategorie a nalézá hledaný produkt.

## **Produkt**

Detail produktu v produktovém katalogu je jednou z nejdůležitějších částí, které bychom při návrhu, podobně jako u informační architektury celého webu, měli věnovat zvýšenou pozornost. Na této stránce jsou zobrazeny podrobné informace o zboží – dostupnost, popis, cena, záruční doby. U pokročilejších e-shopových řešení zde najdeme fotogalerie, videa, uživatelské názory a hodnocení.

Důležité prvky v detailu:

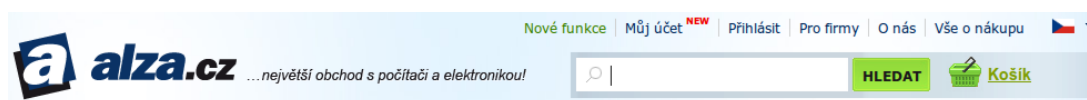
- unikátní název, seriové číslo
- cena (s DPH/ bez DPH)
- popis, atributy produktu
- dostupnost na skladu, dodací lhůta, záruční podmínky
- výzva k akci (většinou realizované tlačítkem „přidat do košíku“)

## Interní vyhledávání a třídění

Pokud pro vlastní řešení nepostačí roztrídění zboží do kategorií je třeba poskytnout zákazníkovi další prostředky k dosažení cíle – pro jednoduché hledání můžeme realizovat fulltextové vyhledávání v našem e-shopu, kdy se prohledávají názvy produktů a jejich popis, uživatel jednoduše zadá hledaný název nebo úryvek do políčka vyhledávání. Ve složitějších případech můžeme naši aplikaci rozšířit o vyhledávání dle atributů – např. podle ceny nebo různých rozměrů, vlastností, váhy, barvy, apod.

## Dopňková navigace

Vedle hlavní navigace se na rozsáhlejších portálech používá doplňující prvek, který se dá nazvat vedlejší či doplňková navigace. Poskytuje uživateli rychlé odkazy do dalších podstránek a funkcí e-shopu, které nejsou součástí produktového katalogu. Nejčastěji se může jednat o reklamační podmínky, aktuality přepnutí do jiného jazyka nebo odkaz do uživatelských nastavení.



Obrázek 13: Doplňková navigace a vyhledávání

Zdroj: (32)

## Košík

Důležitou součástí každého elektronického obchodu je nákupní košík. Narozdíl od svého předchůdce z reálného světa tento nemá kolečka ani rukojeti, oplývá mnohem lepšími funkcemi. Měl obsahovat funkce jako je počítání celkové ceny objednávaného zboží – uživatel ví v každém okamžiku, kolik ho bude nákup stát. Jednoduché přidání zboží by mělo mít ve funkcích zastoupeného svého antagonistu – odebrání konkrétní položky nebo vysypání celého košíku. Pokročilejší internetové obchody mají možnost uložení košíku pro pozdější nákup, přednastavení uživatelských košíků s oblíbeným zbožím pro opakované transakce.

## **Objednávky**

Pokud se zákazník dostane do fáze, že již má všechno zboží v košíku a nepřeje si nic dalšího objednat, přichází krok potvrzení objednávky. V internetovém obchodě je celý proces objednávky realizován několika kroky:

- uplatnění dárkových či slevových kupónů
- zadání fakturačních a dodacích údajů
- výběr dopravce
- zadání metody platby
- potvrzení objednávky

## **Uživatelská nastavení**

Je-li jednou uživatel registrován do systému, měli by mu z toho vyplynout nějaké výhody, pokud ne, měli bychom zvážit přidání funkce - „nakupovat jako host“ bez potřebné registrace. Pro usnadnění nákupu registrovaného uživatele můžeme využít ukládání jeho adres, oblíbených produktů, atd. Přihlásí-li se do systému, měl by mít přehled o uskutečněných i aktuálních objednávkách a fázích, v jakých se zrovna objednávka nachází např. - zpracovává se, balení, k vyzvednutí či na cestě.

### **3.1.2 Backend**

Nedílnou součástí moderních webových aplikací je část backend. Jedná se o administrační rozhraní pro úpravu obsahu a dalšího. Backend poskytuje celou řadu funkcí, pomocí kterých se dá uskutečňovat elektronické obchodování z pohledu majitele obchodu – tato funkcionalita se týká hlavně správy obchodu a jeho modulů. Jedná se hlavně o:

- správa katalogu zboží
- skladové hospodářství
- správa zákazníků
- evidence objednávek

- uprava informačních stránek obchodu - CMS (reklamační řád, obchodní podmínky, adresy), aktualit
- doplňkové marketingové funkce

Přístup do backendu je umožněn pouze určeným uživatelům – majiteli a jeho personálu či administrátorovi obchodu. Děje se tak po ověření hesla a uživatelského jména proti databázi a ověření práv a identity uživatele.

### **Správa katalogu zboží**

Správou katalogu se rozumí celá řada funkcí a možností pro řešení problematiky správy a prodeje zboží. Ne vše, co nastaví administrátor na backendu, lze vidět na frontendu - Může se jednat o nastavení neviditelnosti výrobku, zobrazení až od určitého data.

U každého produktu by měl mít obchodník možnost nastavení ceny, skladového množství, popisu, zkratka všechny možnosti, které chceme zobrazit koncovému uživateli na frontendu. Propojení produktu v rámci skladových programů už je záležitostí pokročilejších řešení, v normálním případě postačí manuálně nastavit skladové množství zboží.

Každý produkt náleží do jedné či více kategorií, lze nastavovat atributy výrobku, fotogalerii. Optimalizační nástroj – spřátelené produkty se dá použít k provázání celého obchodu a usnadnění navigace, vedlejší účinek bude růst pozic ve vyhledávání.

Speciální cenové akce a poštovné se samozřejmě hodí zobrazovat na titulní straně či u konkrétního výrobku.

### **Správa zákazníků**

Všichni registrovaní zákazníci by měli mít bezpečně uložené přihlašovací údaje v databázi obchodu i se svými nastaveními. Obchodník má možnost, dovoluje-li mu to rozhraní, dělit zákazníky do skupin – každá skupina má určitá privilegia – např. slevy, poštovné zdarma. Dělení je možné na základě počtu objednávek, celkové ceně odebraných výrobků nebo jenom podle stáří registrace (odlišíme věrné zákazníky).

Skupiny členství zákazníků můžeme líbivě pojmenovat např.: zlatí, stříbrní, bronzoví a každé skupině určit interval členství – zmiňovanou cenu, odběr, atd.

Vedle uživatelských skupin by zde měla být možnost smazat uživatelský účet, obnovit heslo či monitorování online uživatelů.

### **Evidence objednávek**

Pro obchodníka je velice důležité, aby rozhraní backendu bylo přehledné a správně uspořádané. To samé platí i správy objednávek, jelikož přehlédnutí objednávky a její nevyřízení rovná se ztrátě na zisku a ohrožení samotného přežití obchodu.

Systém by měl povolit změnu stavu objednávky obchodníkem – např. „na skladě“, „na cestě“, „připravena k vyzvednutí“. Automatické vygenerování faktury z obsahu košíku je v dnešních systémech samozřejmostí. Vymazání či zrušení objednávky musí být pro obchodníka k dispozici v případě chyby na straně zákazníka. Možnosti dělat si pokročilé reporty napříč objednávkami, časem a tržbami zahrnují velice silný reportovací nástroj, jelikož podle výkazů může majitel plánovat rozvoj či vylepšení obchodu a služeb.

### **Správa dodání a dopravců, platebních systémů**

Každá objednávka se musí nějakým způsobem dostat k zákazníkovi do rukou. V ČR je velice preferováno osobní převzetí na pobočce. Firma v podstatě umožní zákazníkovi za cenu menší než při koupi v kamenném obchodě objednat zboží a tím podporuje svůj internetový prodej, šetří se tím náklady obchodníka i zákazníka, přičemž je možné zboží připravit a poté u zvláštní přepážky vyzvednout bez asistence prodejního specialisty, který může vyřizovat návštěvníky pobočky, kteří potřebují vysvětlení k výrobkům.

Internetový obchod by měl obsahovat možnosti nastavení ceny dopravců, dodací podmínky a počítání celkového poštovného např. podle váhy, velikosti a smluvních podmínek s konkrétním dopravcem.

V dnešní době probíhá většina platebního styku přes internet. Vedle platby v hotovosti, na dobírku a bankovním převodem máme v dnešní době celou řadu implementovaných možností, jak zpříjemnit zákazníkovi platbu – pokud má samozřejmě kreditní kartu.

Platba kartou patří k ožehavým záležitostem z hlediska bezpečnosti, ovšem banky již mají řešení v podobě zabezpečených bran 3D SECURE, které poskytují akceptaci platebních karet VISA a MASTER CARD.

## 3.2 Dostupné možnosti při realizaci e-shopu

Webdeveloperská firma, která nabízí zákazníkům realizaci internetového obchodu, má na výběr z několika modelů realizace. Jsou to zejména tyto:

- open source řešení
- obchod na míru
- prodej nebo pronájem hotového řešení

V této práci se budu zabývat analýzou otevřených řešení a také projektem na míru. V současné době to jsou pro mě jediná dostupná řešení, která mohu zákazníkovi nabídnout. Pro prodej nebo pronájem krabicového řešení bych musel mít funkční základ s nastavovatelnými moduly, v dalším textu se proto těmito alternativami nebudu zabývat.

Každé řešení má svoje pro a proti. Faktorů, které musíme vzít v případě takovýchto modelů prodeje či pronájmu, je několik. Zaměřujeme se na následující:

- cena
- doba uvedení do chodu
- požadovaná funkcionalita
- rozšiřitelnost
- technická podpora

Níže uvedené časové kalkulace pro jednotlivé typy projektů se budou zakládat na mých zkušenostech z již realizovaných zakázek a pokusných instalací. Tyto údaje se mohou projekt od projektu lišit, budu se tedy snažit zvolit jakýsi obecný výpočet jednotlivých časových úseků za předpokladu, že se při implementaci nevyskytnou žádné závažné a těžko řešitelné problémy, které by mohly celý interval prodloužit.

### 3.2.1 Open source řešení

Open source řešení mohou být vyvíjena celou komunitou nebo firmou či jednotlivci po celém světě. Tato licence otevírá zdrojové kódy široké veřejnosti. Nádledně takto uvolněný software mohou programátoři upravovat. Mohou např. používat základní řešení pro svoje projekty a toto jádro mohou rozšiřovat o moduly, které si nechají

zaplatit. Dalším modelem komerčních firem může být např. to, že si nechají platit za instalace tohoto softwaru, konzultace či úpravu některých součástí. Konkrétně je třeba zmínit robustní řešení **Magento** a oproti tomu rozsahově menší **Prestashop**.

## Magento

Firma, která dnes vyvíjí Magento byla založena v roce 2001. V té době ještě nebylo zcela zřejmé, že se tento program stane jejich hlavním artiklem. Postupně se od menších webových aplikací, jako jsou menší redakční systémy, a intranetů přeorientovala na open source řešení. Myšlenka, že vyvinou takto otevřený systém, je napadla v roce 2007 a v té době už byly v této komunitě, která se shlukla kolem nápadu, tisíce vývojářů. V roce 2008 byla vydána verze 1.0 a začala se datovat éra tohoto programu. Pro náročnější zákazníky, které neuspokojila funkcionality otevřené verze, firma poskytla model Enterprise edice v roce 2009, pořídit se dá za několik tisíc dolarů (33).

The screenshot displays the Magento Admin Panel's 'Objednávky' (Orders) section. At the top, there's a navigation bar with 'Prodejce' (Seller) selected. Below it, a search bar and a message bar are visible. The main content area shows a table of orders. The table has columns for 'Objednávka č.' (Order number), 'Koupeno na' (Purchased on), 'Účet na jméno' (Account name), 'Zásilka na jméno' (Shipping name), 'Celkový součet (základ)' (Base total), 'Celkový součet (zakoupený)' (Purchased total), 'Stav' (Status), and 'Událost' (Action). Two orders are listed, both with a status of 'Pending' and a total of 408,00 Kč.

Objednávka č.	Koupeno na	Účet na jméno	Zásilka na jméno	Celkový součet (základ)	Celkový součet (zakoupený)	Stav	Událost
100000002	22.10.2012 6:33:09	David Stránský	David Stránský	408,00 Kč	408,00 Kč	Pending	Pohled
100000001	22.10.2012 6:03:03	David Stránský	David Stránský	408,00 Kč	408,00 Kč	Pending	Pohled

Obrázek 14: Výchozí administrační rozhraní Magento  
Zdroj: (vlastní instalace)

S důrazem na otevřenost řešení je celý obchod naprogramován v interpretovaném jazyce PHP 5, konkrétně se používá framework Zend s architekturou MVC. Databázovou vrstvu zajišťuje systém MySQL, vlastněný firmou ORACLE. Celé řešení může pohánět webový server Apache nebo jiný. Celý systém se dá snadno rozšířit o bezplatné moduly či komerční moduly pomocí systému Magento Connect. V současné době je volně ke stažení komunitní verze 1.7.0.2, vydána jako stabilní. Ke stažení se stačí pouze na serveru zaregistrovat. U nás se zabývá vývojem modulů a poskytováním hostingu pro Magento firma NoStress Commerce, s.r.o.



## **Výhody**

- ✓ množství pokročilých funkcí v základní instalaci
- ✓ komunitní verze zdarma
- ✓ snadná rozšiřitelnost o moduly přes Magento Connect
- ✓ kvalitně naprogramovaný systém ve frameworku Zend
- ✓ lze rozšiřovat vlastní šablony vzhledu
- ✓ komunita vývojářů, která systém stále vyvíjí – častější aktualizace
- ✓ poměrně rychlé uvedení do chodu

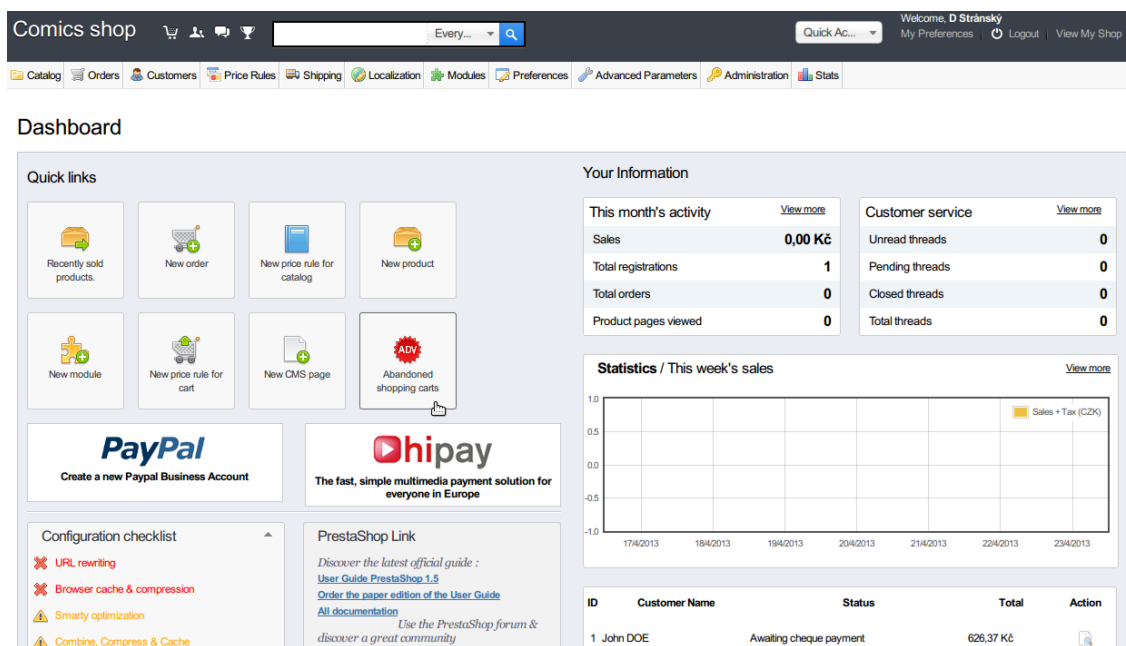
## **Nevýhody**

- ✗ větší nároky na hostingový server
- ✗ omezení vyplývající z toho, že systém je již naprogramován (někdy se těžko upravuje pro specifické požadavky klienta)
- ✗ potřeba důkladné znalosti systému a programování při jeho nasazení a rozšiřování (Zend)
- ✗ občas se vyskytne bezpečnostní či programová chyba (opraveno v další aktualizaci)

## **Prestashop**

Open source řešení Prestashop je vyvíjeno od května roku 2007, původně bylo přeloženo do 13 světových jazyků, v dnešní době je ale systém lokalizován do 56 jazyků. Podle statistik vývojářů existuje okolo 130 000 živých instalací tohoto řešení a denně si instalační soubory stáhne 3500 zájemců. Prestashop je také držitelem ocenění pro nejlepší obchodní open source aplikaci roku 2011(34).

Vlastní systém je naprogramován pomocí OOP skriptovacího jazyka PHP 5, bez žádného frameworku třetích stran, vývojáři tak vyšší rychlosti aplikace a menší náročnosti na hosting. Celý systém používá jednoduchou třívrstvou architekturu, stejný princip jako MVC, jen poněkud jednodušší (35).



Obrázek 15: Administrační rozhraní Prestashopu

Zdroj: (vlastní instalace)

Systém je rozšiřitelný pomocí doplňkových modulů, tzv. addonů. Stejně jako u Magenta některé lepší moduly jsou placené, ale stále je jich spousta dostupná zdarma. V současné době je dostupný Prestashop verze 1.5.4.

### Výhody

- ✓ běžné požadavky na hosting
- ✓ instalaci podle návodu zvládne i laik
- ✓ snadno stažitelné moduly a témata vzhledu
- ✓ rozšiřitelnost jenom se znalostí technologií PHP, MySQL, není třeba se učit nový framework
- ✓ obsahuje všechny běžné marketingové nástroje

### Nevýhody

- ✗ hodí se pro středně velké projekty
- ✗ některé potřebné moduly je třeba navíc zaplatit
- ✗ pokud neexistuje dostupný modul, tak se systém těžko přizpůsobuje zákazníkovi na míru
- ✗ pro pokročilejší úpravu je třeba konzultace a programování webového vývojáře

## Postup při realizaci

Celý proces nasazení open-source systému internetového obchodu se dá rozdělit na několik kroků.

1. zjištění potřeb zákazníka
2. definování cílů a funkcí aplikace
3. instalace otevřeného řešení
4. nastavení dostupných funkcí
5. doinstalace (zakoupení, stažení volně dostupných) nebo doprogramování rozšiřujících modulů
6. tvorba šablony vzhledu (z dodaných grafických podkladů)
7. plnění obchodu obsahem (výrobky, kategorie, podstránky)
8. provozování obchodu

## Časová kalkulace

Činnost	Čas [h]
<b>konzultace se zákazníkem a stanovení funkcionality</b>	10
<b>instalace otevřeného řešení</b>	5
<b>nastavení dostupných funkcí</b>	10
<b>instalace dostupných modulů</b>	10
<b>tvorba šablony vzhledu</b>	15
<b>plnění obchodu obsahem</b>	15
	65

Tabulka 1: Časová kalkulace implementace open-source obchodu.

Zdroj: (vlastní zpracování)

V časové kalkulaci pro implementaci otevřeného řešení viz Tabulka 1, lze pozorovat intervaly, které stráví webdesigner na jednotlivých úkolech. Zahrnul jsem zde i instalaci dostupných modulů, které pokud se hodí k našemu řešení, je možné nasadit – samozřejmě, když jsou placené, je třeba zaplatit vývojáři daného modulu, v případě tzv. volných modulů nic neplatíme. Nezahrnul jsem zde časový odhad doprogramování modulu přesně na míru, jelikož se jeho funkcionality a tudíž i náročnost na jeho implementaci může rapidně lišit podle konkrétního požadavku - např. relativně jednoduchý modul generování faktury a rozsahově složitá implementace ERP aplikace do obchodu Magento.

### 3.2.2 Obchod na míru

Realizací obchodu na míru se budu zevrubně zabývat v praktické části práce. Tato kapitola bude věnována analýze postupu implementace řešení a kalkulace časové náročnosti.

#### Postup při realizaci

Proces tvorby obchodu na míru je od aplikace otevřeného řešení trochu komplikovanější. Nicméně zůstávají některé společné kroky.

1. zjištění potřeb zákazníka
2. definování cílů a funkcí aplikace
3. návrh
  - definování případů užití
  - návrh databázového modelu
4. implementace aplikace
  - zvolení technologií
  - definování architektury programu
  - programování za použití zvolených technologií
  - testování
5. instalace a nastavení systému
6. tvorba šablony vzhledu
7. naplnění systému obsahem
8. provozování a zkušební chod obchodu

## Časová kalkulace

Činnost	Čas [h]
<b>konzultace se zákazníkem a stanovení funkcionality</b>	10
<b>návrh aplikace</b>	
definování případů užití	10
návrh databázového modelu	20
<b>implementace aplikace</b>	
zvolení technologií	4
definování architektury programu	10
programování za použití zvolených technologií	100
testování	10
<b>instalace a nastavení systému</b>	15
<b>tvorba šablony vzhledu</b>	15
<b>naplnění systému obsahem</b>	15
	209

Tabulka 2: Časové náklady realizace obchodu na míru

Zdroj: (vlastní zpracování)

Jak bylo řečeno, jednotlivé kroky jsou částečně shodné s realizací otevřeného obchodu. Není zde zahrnuta tvorba dodatečných funkcionalit, které také mohou být velice specifické. Např. pokud zákazník bude chtít za rok rozšířit obchod o další platební systémy. Zde je situace výhodnější pro rozšíření, jelikož systém není vázán předpřipravenou komunitní architekturou a strukturou programu. Jak vidíme, časová náročnost realizace je odhadnuta asi třikrát vyšší než u předcházející alternativy. Tato investice se ale zákazníkovi vrátí při výše zmíněném rozšiřování. Jelikož budu znát kompletně celou aplikaci, budu si smět dovolit nabídnout nižší sazbu za vylepšení funkcí.

### **3.3 Analýza současné situace firmy**

#### **3.3.1 Popis firmy**

Internetový obchod *Polovnaoptika.sk*, je vlastněn firmou MV Trade – jednatel Miroslav Václavík (36).

**IČ DPH:** SK1048218325

**IČO:** 44929846

Celý obchod je koncipován s ohledem na prodej mysliveckého zboží.

#### **Vybrané kategorie zboží:**

- noční vidění
- zaměřovače
- dalekohledy
- termovize
- nože, fotopasti, instruktážní videa, atd.

V zájmu klienta nebudu v této práci uvádět žádná konkrétní data a firemní statistiky. Na základě konzultace ale vypracuji zjednodušenou SWOT analýzu, podle které naplánuji další kroky.

#### **3.3.2 SWOT analýza**

SWOT analýza se používá k odhalení silných a slabých stránek (interterních faktorů), příležitostí a hrozeb (externích faktorů) naší firmy. Tato metoda nás důkladně donutí zamyslet se nad faktory, které ovlivňují naše podnikání (37).

#### **Interní analýza**

V interní analýze je třeba vyhodnotit slabé a silné stránky, viz Tabulka 3 a Tabulka 4. Každému kritériu přidáme váhu, která bude procentuální vzhledem k ke všem kritériím, součet vah by měl dát 100%. Následně každé konkrétní kritérium ohodnotíme v případě silných stránek kladně a u slabých stránek záporně.

Silné stránky			
Kritérium	Váha	Ohodnocení	Výsledek
Kvalitní výrobky	0,3	4	1,2
Krátká dodací lhůta	0,2	3	0,6
Dobré marže	0,3	3	0,9
Věrní zákazníci	0,1	3	0,3
Kvalifikovaný personál	0,1	5	0,5
Součet			3,5

Tabulka 3: Silné stránky SWOT

Zdroj: (vlastní zpracování dle konzultace)

Slabé stránky			
Kritérium	Váha	Ohodnocení	Výsledek
Nízký rozpočet na reklamu	0,2	-3	-0,6
Malý konverzní poměr e-shopu	0,4	-5	-2
Závislost na dodavateli	0,1	-2	-0,2
Zastaralý design obchodu	0,2	-4	-0,8
Málo jazykových mutací	0,1	-1	-0,1
Součet			-3,7

Tabulka 4: Slabé stránky SWOT

Zdroj: (vlastní zpracování dle konzultace)

Po výše uvedených krocích nám u každého kritéria vynásobením ohodnocení a váhy vyjde výsledek, který udává důležitost kritéria a jeho současný stav. Součet všech výsledků použijeme pro vyhodnocení interních faktorů.

$$S (\text{silné stránky}) + W (\text{slabé stránky}) = 3,5 + (-3,7) = -0,2$$

### Analýza externích faktorů

Příležitosti			
Kritérium	Váha	Ohodnocení	Výsledek
Posílení internetového marketingu	0,2	2	0,4
Spolupráce s novými dodavateli	0,1	4	0,4
Implementace platební brány	0,3	1	0,3
Slevové akce	0,1	2	0,2
Rozšíření sortimentu	0,2	3	0,6
Součet			1,9

Tabulka 5: Příležitosti SWOT

Zdroj: (vlastní zpracování dle konzultace)

Hrozby			
Kritérium	Váha	Ohodnocení	Výsledek
Změna algoritmu vyhledavače	0,3	-3	-0,9
Legislativa ohledně zbraní	0,3	-5	-1,5
Vstup nového konkurenta	0,15	-2	-0,3
Zdražení PPC reklamy	0,2	-3	-0,6
Zvýšení daní	0,05	-2	-0,1
Součet			-3,4

Tabulka 6: Hrozby SWOT

Zdroj: (vlastní zpracování dle konzultace)

Kroky, které jsme uplatnili u analýzy interních faktorů užijeme i zde. Sečteme příležitosti i hrozby.

$$O \text{ (příležitosti)} + T \text{ (hrozby)} = 1,9 + (-3,4) = -1,5$$

Interní a externí faktory dáme do bilance.

$$I + E = -0,2 + (-1,5) = -1,7$$

Pokud máme tuto bilanci zápornou musíme zapracovat na zlepšení. Nejvíce se nám podaří ovlivnit některé interní faktory.

### Závěr SWOT

Z uvedené záporné bilance je patrné, že je třeba podniknout nějaké kroky k vylepšení současné situace. S majitelem e-shopu jsme se dohodli na vytvoření modulu nového internetového obchodu. Vzhledem k finanční situaci bude nový obchod poskytovat základní funkce a až si na sebe vydělá bude docházet k jeho vylepšování. Podmínkou je tedy snadná rozšiřitelnost.



## 4 NÁVRH ŘEŠENÍ

### 4.1 Požadavky na aplikaci

Před návrhem a samotnou realizací aplikace je třeba si vyjasnit specifikaci požadavků. Ze závěru SWOT analýzy nám vyšlo, že nejsnadnější krok k vylepšení současných poměrů bude vytvoření nového internetového obchodu. S majitelem firmy jsme se dohodli, že nepoužijeme open-source řešení, ale v rámci přidání nových funkcí v budoucnosti bude realizován základní modul obchodu na míru. Po čase se vždy obchod rozšíří o novou funkcionalitu podle přání klienta. Zpočátku tak bude řešení poněkud dražší, což se vrátí v budoucnosti po realizaci rozšiřujících modulů.

#### Základní implementované funkce

- administrátorská a veřejná část systému
- správa uživatelů
- produktový katalog
- nákupní košík
- správa objednávek
- nový design (*nebude součástí této práce*)

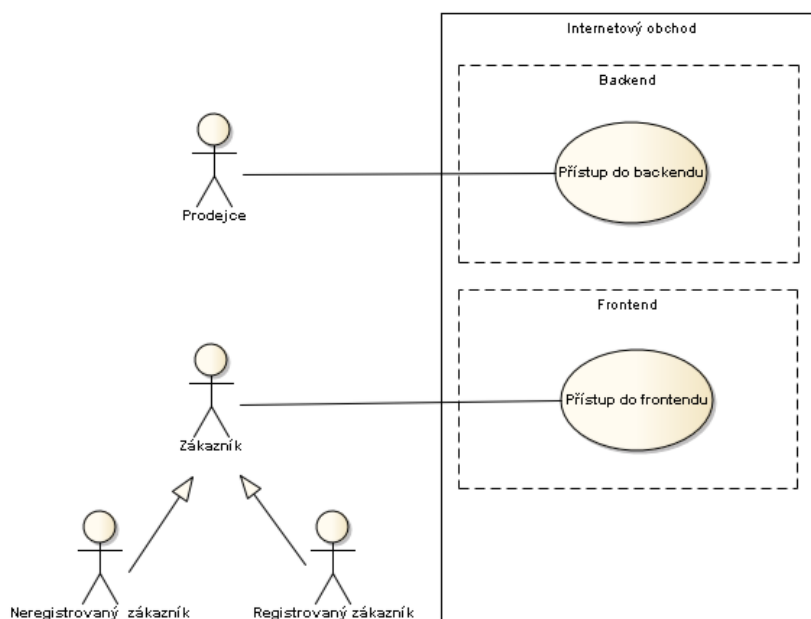
#### Funkce pro implementaci v budoucnosti

- jazykové mutace
- slevové akce
- platební brány
- možnost konfigurovatelného výrobku
- zabezpečení pomocí certifikátů
- správa skladových zásob
- export do zbožíových srovnávačů

Cílem této práce bude navrhout základní modul systému internetového obchodu, který bude možné do budoucna snadno rozšířit. Případy užití aplikace si popíšeme v následující kapitole.

## 4.2 Návrh aplikace

### 4.2.1 Aplikace internetového obchodu a uživatelé



Obrázek 16: Hlavní části systému a uživatelské role

Zdroj: (vlastní zpracování)

V analytické části práce bylo řečeno, že většina složitějších aplikací je rozdělena na frontend a backend. Tento model dodržím i ve své aplikaci. V rámci systému je určeno několik rolí uživatelů. Každá role má přístup do specifické části systému.

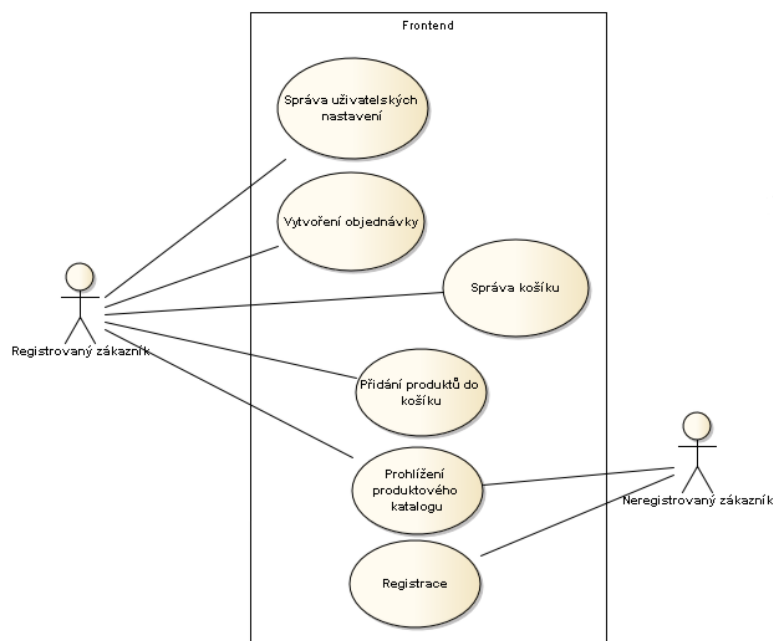
Konkrétně tedy můžeme výjmenovat základní role:

- prodejce
- zákazník
  - registrovaný
  - neregistrovaný

#### 4.2.2 Frontend - případy užití

Neregistrovaný zákazník má možnost se pohybovat ve veřejných částech aplikace, jako je např. produktový katalog, statické stránky (obchodní podmínky, reklamační řád, kontakty). Pokud chce dokončit proces objednávky, musí se zaregistrovat do systému pomocí registračního formuláře, kam zadá e-mail, který se stane jeho přihlašovacím

jménem a heslo. Po registraci do systému se stane uživatel registrovaným – otevírá se mu širší funkcionality frontendu.



Obrázek 17: Use case diagram frontendu

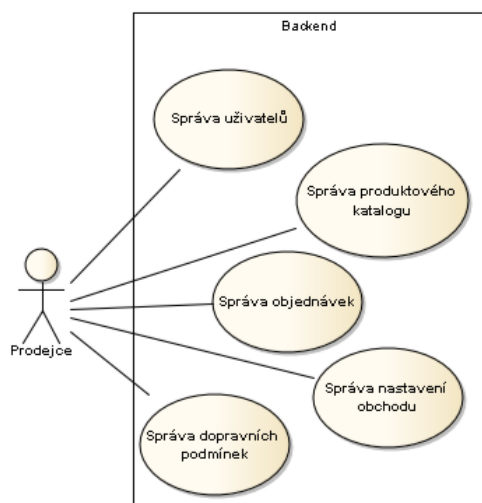
Zdroj: (vlastní zpracování)

Registrovaný zákazník, který se přihlásil do systému může používat části frontendu, kam neměl neregistrovaný přístup, viz Obrázek 17. Jedná se hlavně o správu uživatelských detailů: uložení dodacích a fakturačních adres, přehled o objednávkách. Může také přidávat zboží svého do košíku a košíky si ukládat pro pozdější použití. Pokud má zboží vybráno, může přikročit k vytvoření objednávky. Daný proces objednávky zboží si detailněji popíšeme později.

### 4.2.3 Backend – případy užití

Do backendu aplikace má přístup pouze prodejce – majitel internetového obchodu. Našem případě si vystačíme pouze s jedním typem administrátora. Pokud by aplikace byla rozsáhlejší a bylo třeba rozlišit více druhů administrátorských účtů, přidali bychom role jako je např. zaměstnanec, vlastník obchodu, plnič výrobků, atd. Každá tato role by měla ve frontendu svoje práva a mohla např. přidávat výrobky nebo spravovat objednávky atd. Ověření uživatelových práv by se dělo v rámci každého modulu.

Jak bylo řečeno, máme jenom jeden typ administrátora, který po autentizaci a autorizaci v systému, má na výběr z funkcionality části pro správu.



Obrázek 18: Use case diagram backendu

Zdroj: (vlastní zpracování)

První administrátorský účet je vytvořen programátorem obchodu. Administrátor poté může vytvářet další administrátorské účty.

Prodejce (administrátor) má k dispozici funkce pro správu obchodu viz Obrázek 18.

Podrobněji si tuto funkcionalitu můžeme rozepsat:

- správa uživatelů – mazání, přidávání, úprava detailů
- správa katalogu – mazání, přidávání, úprava kategorií, produktů (variant)
- správa objednávek – vyřizování objednávek, případně zrušení a úprava stavu
- nastavení obchodu – změny obchodních podmínek,
- správa dopravních podmínek, způsobů platby – úprava poštovního a druhů plateb

#### 4.2.4 Popis procesu objednání zboží

V kapitole si popíšeme jeden proces v realizovaném internetovém obchodě. Mohli bychom zde popsat všechny procesy v takovémto druhu informačním systému, ale cílem práce není zpracovat kompletní procesní dokumentaci, a navíc princip zůstává u dalších procesů stejný.

Pro potřeby popisu procesů v informačním systému se používají tři metody vyjádření, jsou to tyto:

- slovní popis
- grafický popis
- tabulkový popis

Pro co nejsrozumitelnější vyjádření by se měly v projektové dokumentaci využívat alespoň dva tyto prostředky (38, str. 49).

U procesu popsaného níže a skutečné implementace programu by mohl být menší rozdíl, funkce přidání do košíku může být dostupná až po přihlášení uživatele do systému z důvodu ochrany proti spamovým robotům, na druhou stranu pro uživatele je lepší přihlášení až ve chvíli, kdy chce dokončit objednávku.

### Grafický popis procesu – EPC diagram



Obrázek 19: EPC diagram objednávkového procesu

Zdroj: (vlastní zpracování)

V tomto EPC diagramu (Obrázek 19) jsem pro zjednodušení v zájmu toho vynechal bloky – nástroje pro informační podporu procesu, jelikož všechny funkce řeší implementovaný obchod. Taktéž pro zjednodušení zkracuji názvy odpovědností za aktivitu:

- I – informed
- R – responsible

### **Slovní popis procesu**

#### **Krok.1 – vyhledání produktu v katalogu**

Prohlízet webové stránky může i nepřihlášený uživatel. V produktovém katalogu buďto pomocí prohlížení kategorií či interního vyhledávání nalezne požadovaný produkt.

#### **Krok. 2 – přidání produktu do nákupního košíku**

Zákazník přidá zboží do košíku a může dále prohledávat katalog a plnit svůj košík jinými produkty, dokud nechce odeslat objednávku.

#### **Krok. 3 – přihlášení či registrace do systému**

Pokud zákazník chce objednat zboží, musí se nejprve přihlásit do systému. V případě, že nemá svůj uživatelský účet, zaregistruje se. Jakmile je takto ověřen, může pokračovat v objednávce. V případě registrace nového uživatele je majitel obchodu informován o novém klientovi.

#### **Krok. 4 – vytvoření objednávky**

Uživatel vytvoří objednávku – zadá fakturační údaje a adresu, zvolí způsoby platby, dopravy a potvrdí zadané údaje, na jejichž správnosti závisí budoucí vyřízení objednávky. V případě úspěšného odeslání je prodejce informován a může přistoupit k vyřízení objednávky. V této části je produkt objednaný proces objednávání skončil.

### **4.2.5 Návrh databázového modelu**

Pro databázový server aplikace bude sloužit relační MySQL, proto jsem použil k návrhu databázového modelu program MySQL Workbench, který je pro tento typ modelování

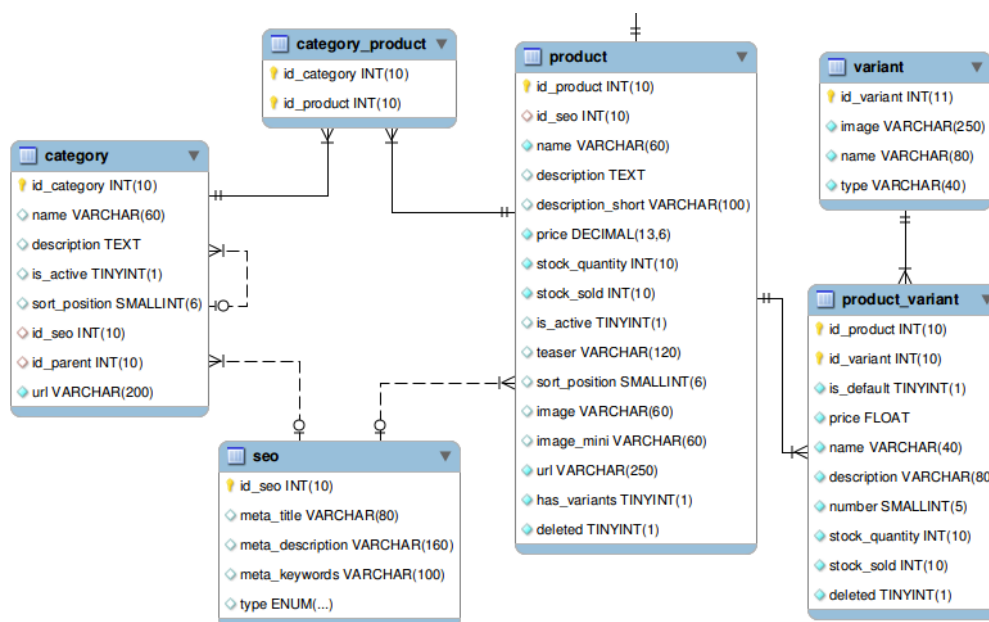
optimální. Celá databáze bude používat kódování UTF-8 (utf\_general\_ci) za použití úložiště dat InnoDB, které podporuje cizí klíče a transakce.

Tabulky v databázi můžeme nazývat entitami, vztahy mezi nimi vazbami. Máme několik druhů vazeb:

- 1:1 – jednomu řádku tabulky odpovídá jeden řádek jiné tabulky
- 1:N – jednomu řádku tabulky odpovídá více řádků v tabulce druhé
- M:N – více řádků z tabulky odpovídá více řádků z druhé tabulky

Kardinalita vztahu vyjadřuje skutečnost, kolik řádků jedné tabulky může vstoupit do vztahu kolika řádku tabulky jiné. Parcialita vyjadřuje povinnost existence vztahu. V návrhu modelu budu využívat tyto vazby - pomohou mi k tomu také primární klíče, které budou definovány v každé tabulce. Referenční integrita bude obstarávána pomocí cizích klíčů. Pro tabulky v databázi budu používat 1. - 3. normální formy, to nám umožní kladení efektivních dotazů na data v nich uložená.

## Produktový katalog



Obrázek 20: ER diagram produktového katalogu  
Zdroj: (vlastní zpracování)

Produktový katalog tvoří několik tabulek, které jsou vzájemně mezi sebou ve vztazích pomocí různých druhů vazeb zmíněných v předchozí stati. Jedná se o část databáze,

která nám umožňuje uložit informace o produktech (variantách) a kategoriích, do kterých náležejí, viz Obrázek 20.

### **Tabulky - product, variant a product\_variant**

V tabulce „product“ jsou uloženy podrobné informace, které se týkají samotného produktu a jeho detailů jako je popis, cena, skladová dostupnost, cesta k ilustračním obrázkům či url adresa, pod kterou bude produkt zobrazen uživateli. Pro smazání produktu je zde atribut „deleted“, který je přítomen z toho důvodu, že když už se produkt bude nacházet v nějaké odeslané objednávce, jeho smazáním bychom porušili referenční integritu databáze. Parametr „is\_active“ nám poslouží pro dočasné zamezení zobrazování produktu v katalogu.

Pokud má produkt více variant, je možné ke každé konkrétní variantě přiřadit obrázek, cenu a skladové množství, každý produkt má určenou výchozí variantu. Více variant může mít vazbu na více produktů. Např. „zaměřovací osa x“ může být k dispozici u více zaměřovačů. A zároveň dalekohled může mít více dostupných os. Tyto varianty jsou navrženy pro potřeby mysliveckého obchodu, u e-shopu s tričky bychom postupovali trochu jinak.

### **Tabulky – category a category\_product**

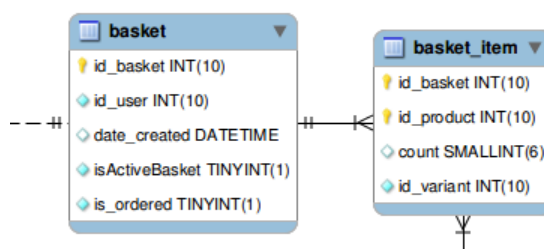
Jak již napovídá název, bude se jednat o tabulky pro kategorizaci produktů. Produkt se bude moci nacházet ve více kategoriích a kategorie může obsahovat více produktů – k tomu slouží vazební tabulka „category\_product“, která umožní vztah M:N. Atribut „is\_active“ v tabulce „category“ umožní dočasné vypnutí zobrazování kategorie na frontendu. Pro potřeby výpisu kategorií ve stromové struktuře slouží vazba sama na sebe pomocí definice cizího klíče „id\_parent“, v tom je vždy uložen nadřazený prvek.

### **Tabulka seo**

Jak již název napovídá bude se jednat o rozšíření pro on-page optimalizaci některých společných prvků kategorie a produktu, jelikož tag meta\_title slouží pro nadpis výsledku v SERPu a meta\_description se zobrazuje někdy ve snippetech úvodníku, u vyhledávače Google taktéž ve výsledcích.



## Nákupní košík



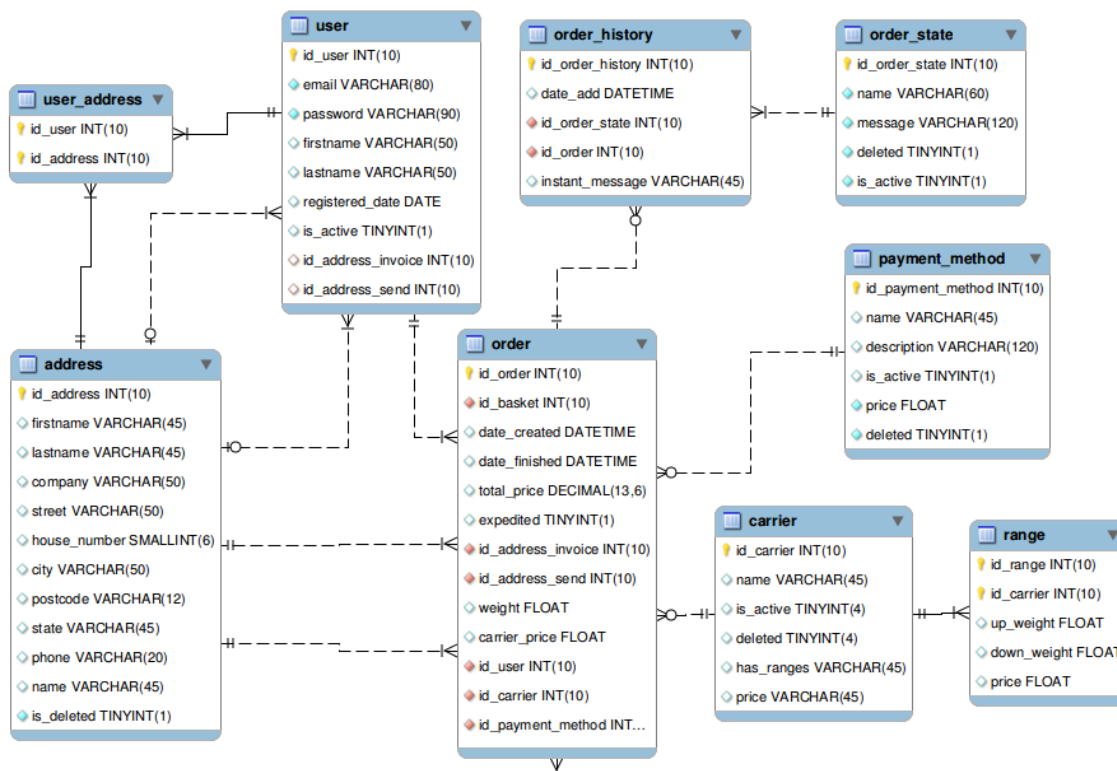
Obrázek 21: ER diagram nákupního košíku  
Zdroj: (vlastní zpracování)

Část nákupního košíku se sestává z tabulky „basket“ a „basket\_item“. Uživatel může mít současně několik uložených košíků, ale jenom jeden aktivní, pomocí kterého nakupuje, a ostatní např. pro pozdější použití. V košíku může být více produktů a každý produkt může být ve více koších, pomocná tabulka „basket\_item“ nám umožňuje realizovat vztah M:N. Položka košíku také umožňuje nastavení množství kupovaného produktu a případně variant.

Košík je po odeslání přiřazen konkrétní objednávce, také se změní atribut „is\_ordered“ na hodnotu TRUE. Poté se již nebude zobrazovat ve výpisu dostupných košíků pro nákup.

## Objednávky a uživatelé

Zde se zastavíme u objednávkového modulu a zákazníků systému. Obě tyto části musí fungovat ve vzájemné kooperaci viz Obrázek 22.



Obrázek 22: E-R Diagram objednávkové části  
Zdroj: (vlastní zpracování)

### Tabulka user

V momentě, kdy se uživatel zaregistruje do systému, vytvoří se mu účet. Pro ukládání údajů o zákazníkovi slouží tabulka „user“. Jsou zde uloženy přihlašovací údaje – uživatelské jméno, heslo. Popřípadě podrobnější atributy jako je jméno a příjmení zákazníka. Pro informativní potřeby je zde také čas registrace, který můžeme později použít pro marketingové účely.

### Tabulka address

Každá objednávka bude mít svoji fakturační a dodací adresu – pro uložení adres slouží tabulka „address“. Přihlášený uživatel bude mít předvoleny svoje výchozí adresy pro snadnější vyplnění objednávky. Také může mít více uložených adres, které použije při častějším zasílání objednávek na různá místa, než jsou výchozí. Jako vazební tabulka zde slouží „user\_address“.

### **Tabulka order**

Po odeslání nákupního košíku se přiřadí k atributu „id\_basket“ tento košík. Obchodník má tak přiřazen unikátní košík k unikátní objednávce. Po odeslání se zde také uloží celková hodnota košíku – atribut „total\_price“, ke které musíme přičíst cenu dopravy - „carrier\_price“. Vyjde nám tak celková cena objednávky.

### **Tabulka carrier**

Každá objednávka musí mít dopravce. Tyto dopravce si definuje administrátor systému a k jejich ukládání slouží tabulka „carrier“. Cenu za dopravu můžeme nastavit fixní nebo jako sofistikovanější řešení lze nastavit ceny podle intervalů váhy zboží (tabulka „range“). Každá objednávka má právě jednoho dopravce a ten může dopravovat více objednávek – pomůže nám vazební tabulka „order\_carrier“.

### **Tabulka payment\_method**

Stejný vztah k objednávce jako tabulka dopravce má také tabulka platebních metod „payment\_method“. Každá objednávka může mít pouze jednu platební metodu.

### **Tabulky order\_state a order\_history**

Stavy objednávek si definuje majitel obchodu – může se jednat o stavy: „na cestě“, „balí se“, „přijata“. Zákazníkovi se poté zobrazí u každé konkrétní objednávky historie či aktuální stav. To je realizováno pomocí změny v tabulce „order\_history“.

## **4.3 Implementace aplikace**

### **4.3.1 Použité technologie**

Když vyvíjíme webovou aplikaci, ať už je to jednoduchá stránka nebo komplikovanější projekt, jako je redakční systém či internetový obchod, máme na výběr z nepřeberného množství vývojářských nástrojů. Můžeme zvolit úplně ty nejjednodušší, u kterých nám stačí základní podpora funkcionality pro zabezpečení chodu projektu – s tím si vystačíme u již zmiňovaných triviálních úkolů. Na druhé straně jsou ovšem trochu sofistikovanější nástroje, které jednak zapouzdřují funkčnost těchto základních prostředků a také přinášejí pro tvorbu aplikací nějakou přidanou hodnotu. Tyto nástroje nazýváme frameworky. Jsou to sady knihoven pro tvorbu, v našem případě webových aplikací, usnadňující základní úkony, které se vyskytují při programování našeho projektu. K realizaci internetového obchodu použiju několik takovýchto frameworků. Nejprve se ale seznámíme se základními nástroji.

#### **LAMP**

Zkratka LAMP je akronym a jeho počáteční písmena znamenají – Linux (operační systém), Apache (webový server), MySQL (databázový server) a PHP (programovací jazyk). Jedná se o sadu technologií, které se používají při vývoji webových aplikací. Při vytváření mé aplikace jsem používal operační systém Ubuntu 12.04.2 LTS (distribuce s dlouhodobou podporou). Hotová aplikace bude také běžet na operačním systému typu Linux, hostována webovým serverem.

#### **Webový server Apache**

K hostování webových aplikací se mimo jiné používá server Apache. Tento server je vyvíjen postupně od roku 1993 velkou skupinou vývojářů po celém světě. Pomocí modulů podporuje celou řadu funkcí, které využíváme pro zabezpečení chodu. Podporuje skriptovací programovací jazyky PHP, PERL a také umožňuje používat zabezpečená připojení, kompresi dat a další. Server můžeme nastavit pomocí konfiguračních souborů, které lze upravovat textovým editorem. Je dostupná i propracovaná dokumentace (39, str. 572-583).

## **Značkovací jazyk xHTML**

Pro odlišení struktury webového dokumentu od jeho vzhledu se používají značkovací jazyky pro hypertext. Mohou rozdělit celý dokument do logických částí – bloků, odstavců, nadpisů. V rámci těchto pak provázat podobné soubory a části pomocí odkazů. Texty formátované pomocí značkovacího jazyka lze zobrazit v internetovém prohlížeči, který je schopen rozlišovat syntaxi. Pro účely mého projektu bude použito verze xHTML 1.0 Transitional.

## **Kaskádové styly CSS**

Tabulky kaskádových stylů CSS umožňují, pokud je máme připojené k HTML dokumentu, definovat jeho vzhled. Je totiž třeba odloučit vzhled dokumentu od jeho formátování a obsahu. CSS přiřazují jednotlivým tagům značkovacího jazyka různé vlastnosti na zobrazovacím zařízení – velikost písma, barvu textu, pozadí, rozměry prvků. Bude použita revize 2.1, která je podporována v širokém spektru současných prohlížečů.

## **Databázový server MySQL**

K uložení aplikačních dat je dostupné řešení databázového systému MySQL. Je také volně ke stažení pod bezplatnou GPL licenci. Tento balík umožňuje vytváření uživatelských databází, definice databázových struktur, plnění daty a provádění operací nad těmito daty a strukturami. Pro vývoj projektu jsem použil dostupnou verzi 5.5.29.

## **Javascritový framework JQuery**

Poslední dobou se v aplikacích stále více používají efekty na straně klienta (prohlížeče), jednak tyto efekty působí dobře na uživatele – máme celou sadu nástrojů pro ověřování uživatelských vstupů, takže nemusí docházet k znovunačtení stránky při špatně zadaných datech, dále můžeme používat různé efekty pro dynamickou úpravu vzhledu - přiřazení různých hodnot kaskádovým stylům, manipulace s elementy v html dokumentu (DOM) a používání asynchronního javascriptového kódu (AJAX), pro načtení pouze malé části stránky. To vše a mnohem více poskytuje javascriptový

framework JQuery, který je volně ke stažení na domovských stránkách projektu pod MIT licenci.

### **CSS Framework Blueprint**

Nejedná se o žádné vylepšení funkcionality samotných kaskádových stylů, ale o vytvoření základních formátovacích pravidel pro realizovaný web. Tato pravidla se zabývají typografií, rozmístěním prvků, resetem a vzhledem prvků dokumentu. CSS framework Blueprint použijí pro usnadnění prvotního formátování a také z toho důvodu, že je součástí základní instalace Yii frameworku.

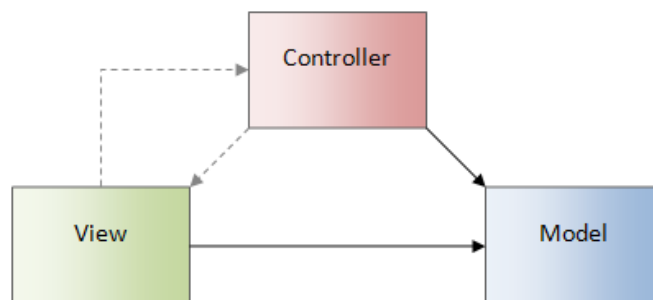
### **PHP framework Yii**

Pro samotnou tvorbu aplikace je třeba všechny zmíněné nástroje a technologie zapouzdřit do jednoho celku a také mít předdefinované knihovny funkcí k často opakovaným programovaným konstrukcím. To přesně nám umožňuje framework Yii (Yes It Is), jehož název je akronymem odpovědi („Ano, to je!“) na otázky: „Je to bezpečné?“, „Je to rychlé?“, „Je to dobré pro můj projekt?“ (40).

Framework je navržen s ohledem na třívtvovou MVC architekturu, o které se zmíníme dále. Také dává k dispozici řadu tříd a funkcí pro práci s uživateli, formuláři, bezpečností či databází.

#### **4.3.2 Architektura aplikace - MVC**

*„Architektura MVC dělí aplikaci na 3 logické části tak, aby je šlo upravovat samostatně a dopad změn byl na ostatní části co nejmenší. Tyto tři části jsou Model, View a Controller. Model reprezentuje data a business logiku aplikace, View zobrazuje uživatelské rozhraní a Controller má na starosti tok událostí v aplikaci a obecně aplikační logiku.“ (41)*

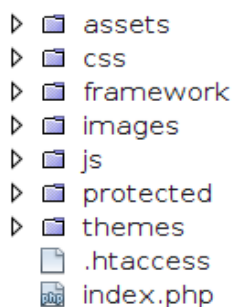


Obrázek 23: Návrhový vzor MVC  
Zdroj: (41)

Pokud se podíváme na Obrázek 23, vidíme základní části architektury MVC, již zmíněný framework Yii, který je použit pro aplikaci, striktně dodržuje. Tento návrh využívá všech dostupných vlastností objekového programování, v samotné aplikaci jsou modely namapovány na databázové tabulky, také mohou být vzájemně provázány. Controller nám vždy pokrývá jednu sadu činností aplikace a dále se dělí na akce, které představují konkrétní činnost a views nám umožňují jednoduše řečeno vypsát výstup konkrétní akce a načíst uživatelský vstup, který pokud projde validačními pravidly modelu je dostupný pro zpracování v contrlleru. Tento popis jsem vzal hodně ze široka a dále si ho podrobněji rozepíšeme, ale nejdříve musíme začít základní adresářovou strukturou, abychom pochopili, kde jsou umístěny základní části.

### 4.3.3 Adresářová struktura a nastavení

#### Kořenový adresář



Obrázek 24: Kořenový adresář aplikace  
Zdroj: (vlastní zpracování)

Kořenový adresář aplikace obsahuje po základní instalaci (předpřipravení kostry), viz Obrázek 24, další podadresáře a soubory. V adresáři **assets** jsou umístěné kódy a doplňující potřebné moduly vygenerované samotným frameworkem (po instalaci

rozšíření třetích stran či vlastních), neměli bychom obsah tohoto adresáře překopírovávat na ostrý server, pokud aplikace zjistí, že nejsou přítomné, znovu se automaticky vygenerují.

Adresář **css** obsahuje definované kaskádové styly frontendu aplikace, pro další vzhledy modulů jsem ještě vytvořil podadresář stylů administrátorské části. Neméně důležitá je také složka **framework**, která jak název napovídá obsahuje všechny knihovny Yii, které nám poskytují většinu funkcí, či znovupoužitelného kódu a také zajišťují chod celé MVC architektury. Adresář **images** je určen pro nahrávání obrázků, které se mohou zobrazovat koncovému uživateli či dále zpracovávat. Část aplikace, kde jsou umístěny další javascriptové knihovny, je adresář **js**. Témata vzhledu mohou být umístěna v **themes**. Všechny výše zmíněné položky jsou samozřejmě důležité, ale kdybych měl zohlednit ty nejdůležitější části kořenového adresáře, musím zmínit soubor **index.php** a adresář **protected**.

### Soubor index.php

```
<?php
$yii=dirname(__FILE__).'./framework/yii.php';
$config=dirname(__FILE__).'./protected/config/main.php';
defined('YII_DEBUG') or define('YII_DEBUG',true);
require_once($yii);
Yii::createWebApplication($config)->run();
```

Výše zmíněný výpis souboru index.php můžeme popsat v pár krocích jako připravení před načtením aplikace – do proměnné \$yii se uloží cesta k hlavnímu souboru frameworku, proměnná \$config obsahuje cestu k naší vlastní konfiguraci aplikace. Definice dalších konstanty (YII\_DEBUG) nám umožňuje spustit ladící režim.. Poté se celý framework načte a pomocí statické metody se aplikace spustí s naší konfigurací.

### Aplikační adresář

Adresář, kde by měly být umístěny všechny programové kódy mé aplikace, je poměrně rozsáhlý, nebudu ho zde tudíž popisovat celý se všemi jeho položkami, pro představu bohatě postačí pouze Obrázek 25. Hlavní části budou podrobněji popsány.





Obrázek 25: Aplikační adresář  
Zdroj: (vlastní zpracování)

U tohoto adresáře je třeba zabezpečit, aby nebyl přístupný z vnějšku aplikace (internetu), o to se stará webový server, musíme ale uvést soubor `.htaccess`, který zakáže nebo povolí přístup do adresáře pomocí direktivy: *deny from all*.

### Nastavení aplikace

V podadresáři **config** se nachází soubor **main.php**, ve kterém se nacházejí všechny globální nastavení aplikace. Jsou zde uvedeny importy modelů, které se automaticky nenačty (nejčastěji z programových podmodulů), nastavení přístupu do databáze, url router a další důležitá nastavení.

```
'db'=>array(
    'connectionString' => 'mysql:host=localhost;dbname=basic-eshop',
    'emulatePrepare' => true,
    'username' => 'root',
    'password' => 'root',
    'charset' => 'utf8',
    'enableProfiling' => true,
    'schemaCachingDuration' => 3600,
),
```

V uvedeném nastavení databáze, kterou jsem používal pro vývoj aplikace, je třeba nastavit správně jméno databáze, uživatelské jméno a heslo. Samozřejmě nesmí chybět kódování databáze. Pro ostrý režim je třeba použít bezpečnější heslo.

### 4.3.4 Programování

V kapitole o architektuře aplikace jsme se seznámili návrhovým vzorem MVC, ten a objektové programování prostopují skrz celou tvorbu mé internetové aplikace. Nejprve se ale zastavíme u generování databázové části aplikace.

#### Databázové tabulky

```
CREATE TABLE IF NOT EXISTS `basket` (  
  `id_basket` int(10) NOT NULL AUTO_INCREMENT,  
  `id_user` int(10) NOT NULL,  
  `date_created` datetime DEFAULT NULL,  
  `isActiveBasket` tinyint(1) NOT NULL,  
  `is_ordered` tinyint(1) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`id_basket`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=57 ;
```

Generování tabulky – jazyk syntaxe MySQL, Zdroj: (vlatní zpracování)

Ve výše zmíněném kódu generování tabulky pro košík můžeme názorně vidět, které datové typy jsou s touto entitou asociovány. Uvedený výpis uvádím pro ilustraci a nebudu zde vypisovat celý soubor pro všechny entity. Podrobnější popis tabulek je možné najít v kapitole o návrhu databáze, v sekci konkrétní tabulky. Všechny tabulky, které máme v diagramu z návrhové části, je možné exportovat pro použití v databázi. Tuto možnost má vývojářský program MySQL Workbench.

#### CRUD generování

Framework Yii disponuje nástrojem pro generování kódu Gii. Používá se pro vytvoření základního kódu modelů, controllerů a views. Pokud potřebujeme rozdělit aplikaci na více funkčních částí použijeme generování tzv. modulu.

#### Model

Pro generování kódu modelů jsem použil výše zmíněný CRUD nástroj. Třída modelu se namapuje na konkrétní entitu databáze a jsou také vytvořeny základní settery a gettery pro všechny veřejné atributy. K zajištění funkcí, které budeme pro aplikaci potřebovat,

musíme dopsat veřejné (public) metody modelu. Příklad modelové třídy košíku jsem uvedl v příloze 1. Modely, které mají být namapovány na databázovou entitu jsou generátorem nastaveny na to, aby dědily z vestavěné třídy frameworku – CactiveRecord, která je potomkem CModel. Při programování poté pracujeme s modelem jako obyčejnou třídou, která je rozšířena o rodičovské funkce a chování.

### Validační pravidla modelu

Pokud potřebujeme ošetřit situaci, kdy např. uživatel zadá do vstupu, který by měl načítat celočíselnou hodnotu, namísto toho řetězec, může dojít k situaci, že databáze vygeneruje chybový kód. Tomuto musíme předejít a zajistit, aby uživatel zadal správná data, popřípadě nedovolit načtení špatných dat do databáze. Pro účely validace uživatelského vstupu máme naštěstí pro třídu **active record** možnost nastavení validátorů pro každý atribut třídy. Slouží k tomu veřejná metoda **rules**, která je pro konkrétního potomka - náš model - přetížena.

### Příklad validačních pravidel pro atribut

```
array('id_user', 'required'),  
array('id_user', 'numerical', 'integerOnly'=>true),
```

### Controller

Třída, která zajišťuje logické propojení mezi modelem (databázovou vrstvou) a views, se nazývá controller. Měl by pokrývat jednu sadu případů užití aplikace – např. vytvořit, upravovat či mazat nákupní košík. Konkrétní controller pro modul objednávky jsem uvedl v příloze 2.

### Akce controlleru

Pro potřeby konkrétního provádění úkolu v rámci controlleru slouží veřejné metody akcí. V akci dochází vlastně k tomu, co přesně dovoluje použití architektury MVC. Nejprve zvolíme jaký model (entitu) chceme zpracovávat, načteme si požadovaný model. Vyhodnotíme data, přenášená v systémových polích (**\$\_GET**, **\$\_POST**, atd.) a na základě definovaných podmínek vykreslíme funkcí **render()** příslušné view. Jména akcí je třeba uvádět v rámci jednoho controlleru unikátní.

## Akce inicializace objednávky

Zdroj: (vlastní zpracování)

```
public function actionInit($id_basket){  
    $basket=Basket::model()->findByPk((int)$id_basket);  
    Yii::app()->session['order'] = array(  
        'id_basket' => $id_basket,  
        'weight'=>$basket->getWeight(),  
        'id_user'=>$basket->id_user,  
        'id_carrier'=>0,  
        'carrier_price'=>0  
    );  
    $this->redirect(array('selectcarrier'));  
    $this->render('init');  
}
```

Zabezpečení akcí se provede pomocí metody *accessRules()* (viz příloha 2), která vrátí pole nastavení pro zakázání, potažmo povolení vykonání akcí konkrétní skupině uživatelů.

## View

V internetových aplikacích většinou potřebujeme zobrazit uživateli výstupní informace, případně načíst vstupní data. K tomu slouží v MVC architektuře část zvaná view. Příklad view výpisu detailu objednávky je v příloze č.3.

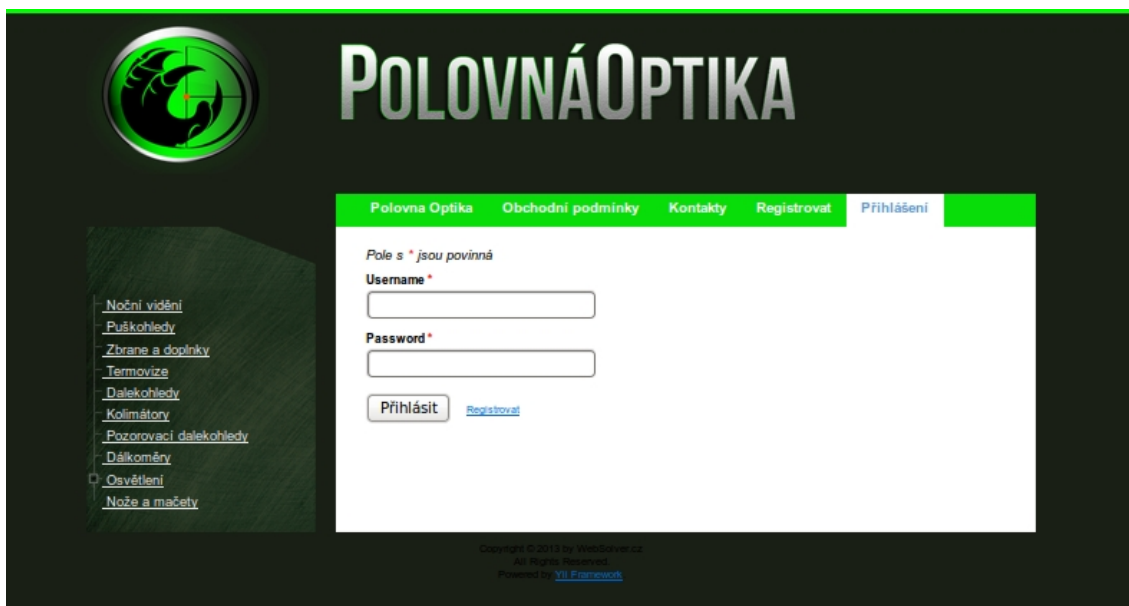
## Moduly

Ve frameworku Yii slouží moduly k oddělení více částí programu od sebe navzájem. Poté každý jednotlivý modul má možnost nastavení svých vlastních výchozích adresářů, tříd a přihlašovacích pravidel. V mém programu jsem vytvořil tyto moduly:

- user
- administrator

Modul user obhospodařuje zákaznickou část aplikace – objednávky, přihlášení uživatele, košíky. Analogicky modul administrator zajišťuje chod backendu.

### 4.3.5 Vzhled aplikace



Obrázek 26: Vzhled naprogramovaného obchodu

Zdroj: (vlastní zpracování dle dodaných grafických podkladů)

Ačkoliv nebylo cílem této práce nakódovat vzhled celé aplikace, uvedu zde příklad přihlašovací části zákazníka. Rozhraní bylo nakódováno (viz Obrázek 26) pomocí kaskádových stylů, z dodaných grafických podkladů (formát psd) od kreativního grafika. V současné době ještě není grafika u některých částí plně implementována.

## ZÁVĚR

V teoretické části jsem si ujasnil východiska pro konkurenceschopnou aplikaci a některé teoretické poznatky, o kterých jsem dosud neměl plné povědomí. I když mám praxi s poskytováním marketingových a programátorských služeb, je tempo růstu v tomto odvětví závratné a klade důraz na vymezení specializace na konkrétní webové „řemeslo“. Mám na mysli to, že si budu muset do budoucna vybrat mezi optimalizací pro vyhledavače, PPC reklamou či programováním webových aplikací. Pokud bych i nadále měl dělat od každého něco, vedlo by to časem k degradaci a absolutnímu neproniknutí k jádru konkrétní problematiky či klouzání po povrchu. Naproti tomu, když se profesně vyprofiluji např. na programování, mám šanci stát se specialistou a proniknout tzv. „pod kapotu“. Tato diplomová práce mi umožnila zjistit potřebu tohoto rozhodnutí.

Myslím, že se mi podařilo naplnit cíle praktické části – návrh a implementace prodejního systému. V kapitole o programování jsem se snažil spíše věnovat pozornost použitým technikám a možnostem frameworku než popisu fungování každé jednotlivé části obchodu, to se nakonec dá v případě zájmu vyčíst z dodaných zdrojových kódů. I když jsem nestihl doprogramovat funkcionalitu pro varianty produktu, stále zůstává podle mě solidní základní aplikace obchodu, která je po dokódování vzhledu a testování schopná reálného chodu. Samozřejmě se budu snažit klienta dotlačit k rozšíření metod plateb a některých dalších specialit, na jejichž popis a implementaci nezbylo v této práci místo. Nicméně mohu říct, že jsem programováním pokryl všechny případy užití aplikace, uvedené v návrhové části. Díky tomu, že jsem použil objektové programování a architekturu MVC, je aplikace připravena pro budoucí rozšíření. Techniky objektového programování používám od začátku vysoké školy, ovšem MVC bylo pro mě trochu větší neznámou. Nakonec jsem tento návrhový vzor, společně s frameworkem Yii, uvedl do praxe pro svoji aplikaci.

## POUŽITÉ ZDROJE

- 1) JANOUGH, Viktor. *Internetový marketing: prosad'te se na webu a sociálních sítích*. Vyd. 1. Brno: Computer Press, 2010, 304 s. ISBN 978-80-251-2795-7.
- 2) BLAŽKOVÁ, Martina. *Jak využít internet v marketingu: krok za krokem k vyšší konkurenceschopnosti*. 1. vyd. Praha: Grada, 2005, 156 s. ISBN 80-247-1095-1.
- 3) ZIKMUND, Martin. Moderní marketing aneb od 4P ke 4C. In: *BusinessVize.cz* [online]. 2010, 2010-01-03 [cit. 2013-05-02]. Dostupné z: <http://www.businessvize.cz/strategie/moderni-marketing-aneb-od-4p-ke-4c>
- 4) KRUTIŠ, Michal. Texty o internetovém marketingu. *Michal Krutiš* [online]. 2007 [cit. 2013-05-02]. Dostupné z: <http://www.krutis.com/texty-o-internetovem-marketingu/>
- 5) KOTLER, Philip. *Moderní marketing: 4. evropské vydání*. 1. vyd. Praha: Grada, 2007, 1041 s. ISBN 978-80-247-1545-2.
- 6) SKOK, David. Understanding the Customer Buying Cycle & Triggers. In: *For Entrepreneurs* [online]. 2012 [cit. 2013-05-02]. Dostupné z: <http://www.forentrepreneurs.com/buying-cycle-and-triggers/>
- 7) TICHÝ, Jan. *Optimalizace pro vyhledávače*. Přednáška. Praha: ČZU, 2011
- 8) PROKOP, Marek. Přidaná hodnota optimalizace pro vyhledávače. In: *Živě.cz* [online]. 2003 [cit. 2013-05-02]. Dostupné z: <http://www.zive.cz/Clanky/Pridana-hodnota-optimalizace-pro-vyhledavace/sc-3-a-114599/>
- 9) HASSMAN, Martin. Marek Prokop: SEO není optimalizace pro vyhledávače. In: *Zdroják* [online]. 2009 [cit. 2013-05-02]. Dostupné z: <http://www.zdrojak.cz/clanky/marek-prokop-seo-neni-optimalizace-pro-vyhledavace/>
- 10) FOX, Vanessa. *Marketing ve věku společnosti Google: využijte ve svém byznysu plný potenciál on-line vyhledávání*. Vyd. 1. Brno: Computer Press, 2011, 245 s. ISBN 978-80-251-3357-6.
- 11) JONES, Jaqui and Daniel SARGEANT, Number of Keywords Within Users' Search Queries. In: *KeywordIntent Blog* [online]. 2010 [cit. 2013-05-02]. Dostupné z: <http://www.keywordintent.com/seo-news/number-keywords-within-searches.html>
- 12) ŘEZNÍČEK, Josef. Jak využít dlouhý ocas (longtail) ve vaší SEO strategii. In: *Včeliště.cz* [online]. 2012 [cit. 2013-05-02]. Dostupné z: <http://www.vceliste.cz/jak-vyuzit-dlouhy-ocas-longtail-ve-vasi-seo-strategii/>
- 13) NĚMEC, Robert. Cílem SEO není růst návštěvnosti, cílem SEO je růst zisku. In: *RobertNemec.com* [online]. 2012 [cit. 2013-05-02]. Dostupné z: <http://seo-sem.robertnemec.com/cil-seo-navstevnost-zisk/>
- 14) Klíčová slova. *Sklik nápověda* [online]. 2013 [cit. 2013-05-02]. Dostupné z: <http://napoveda.sklik.cz/cz/klicova-slova.html>

- 15) *Seznam Sklik* [online]. 2013 [cit. 2013-05-08]. Dostupné z: <http://www.sklik.cz>
- 16) PROKOP, Marek. Kolik klíčových slov se vejde na jednu stránku. In: *Vyhledavace.info* [online]. 2011 [cit. 2013-05-02]. Dostupné z: <http://vyhledavace.info/clanky/245/kolik-klicovych-slov-se-vejde-na-jednu-stranku>
- 17) ENGE, Eric, Stephan SPENCER, Rand FISHKIN, Jessie C STRICCHIOLA a John BATTELLE. *The Art of Seo / Eric Enge, Stephan Spencer, Rand Fishkin and Jessie C. Stricchiola ; foreword by John Battelle*. Sebastopol, Calif.: O'reilly, 2009, xxiv, 574 p. ISBN 05-965-1886-2.
- 18) Ledničky Electrolux - hledat Seznamem. *Vyhledávač Seznam.cz* [online]. 2013 [cit. 2013-05-08]. Dostupné z: <http://search.seznam.cz/?q=lednickyy+electrolux>
- 19) Ledničky Electrolux - Hledat Googlem. *Google CZ* [online]. 2013 [cit. 2013-05-08]. Dostupné z: <http://www.google.cz/search?q=lednickyy+electrolux>
- 20) KUBÍČEK, Michal. *Velký průvodce SEO: jak dosáhnout nejlepších pozic ve vyhledávacích*. Vyd. 1. Brno: Computer Press, 2008, 318 s. ISBN 978-80-251-2195-5.
- 21) SMÍČKA, Radim. *Optimalizace pro vyhledávače - SEO: jak zvýšit návštěvnost webu*. Vyd. 1. Kralice na Hané: Zásilkové knihkupectví J. Smičkové, 2004, 126 s. ISBN 80-239-2961-5.
- 22) THUROW, Shari. Key Information Architecture Concepts Every SEO Should Know. In: *Key Information Architecture Concepts Every SEO Should Know* [online]. 2010 [cit. 2013-05-02]. Dostupné z: <http://searchengineland.com/key-information-architecture-concepts-every-seo-should-know-45288>
- 23) SMITH, Hannah. Information Architecture, Faceted Navigation & Duplicate Content (Oh My!). In: *Information Architecture, Faceted Navigation & Duplicate Content (Oh My!)* [online]. 2011 [cit. 2013-05-02]. Dostupné z: <http://www.seomoz.org/blog/information-architecture-faceted-navigation-duplicate-content-oh-my>
- 24) PROKOP, Marek. 6 základních stylů navigace. In: PROKOP, Marek. *6 základních stylů navigace* [online]. 2001, 2001-12-09 [cit. 2013-05-02]. Dostupné z: <http://www.sovavsiti.cz/c01331.html>
- 25) KRUG, Steve. *Nenuťte uživatele přemýšlet!: praktický průvodce testováním a opravou chyb použitelnosti webu*. Vyd. 1. Brno: Computer Press, 2010, 165 s. ISBN 978-80-251-2923-4.
- 26) DENTON, William. How to Make a Faceted Classification and Put It On the Web. In: *How to Make a Faceted Classification and Put It On the Web* [online]. 2009, 2011-02-01 [cit. 2013-05-02]. Dostupné z: <http://www.miskatonic.org/library/facet-web-howto.html>



- 27) STOJANOV, Petr. SEO aneb technické překážky indexace. In: *SEO aneb technické překážky indexace* [online]. 2010 [cit. 2013-05-04]. Dostupné z: <http://www.webninja.cz/clanek/17-seo-aneb-technicke-prekazky-indexace>
- 28) TICHÝ, Jan. Ideální odkaz. In: *Medio Blog* [online]. 2012 [cit. 2013-05-02]. Dostupné z: <http://blog.medio.cz/idealni-odkaz>
- 29) DVOŘÁK, Jiří a Jiří DVOŘÁK. *Elektronický obchod: studijní text pro kombinované studium*. Vyd. 1. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2004, 78 s. ISBN 80-214-2600-4.
- 30) ŠVADLENKA, Libor a Radovan MADLEŇÁK. *Elektronické obchodování*. Vyd. 1. Pardubice: Institut Jana Pernera, 2007, 163 s. ISBN 978-80-86530-40-6.
- 31) Nová registrace. *CZC.cz - počítače a elektronika* [online]. 2013 [cit. 2013-05-18]. Dostupné z: <https://www.czc.cz/registrace/nova>
- 32) *Alza.cz - největší obchod s elektronikou* [online]. 2013 [cit. 2013-05-18]. Dostupné z: <http://www.alza.cz/>
- 33) A History of Magento. In: *Orange Collar Media* [online]. 2011 [cit. 2013-05-02]. Dostupné z: <http://orangecollarmedia.com/a-history-of-magento>
- 34) About PrestaShop. *PrestaShop.com* [online]. 2013 [cit. 2013-05-02]. Dostupné z: <http://www.prestashop.com/en/about-us>
- 35) Fundamentals - PrestaShop 1.5. *PrestaShop.com* [online]. 2012, 2013-01-17 [cit. 2013-05-18]. Dostupné z: <http://doc.prestashop.com/display/PS15/Fundamentals>
- 36) VÁCLAVÍK, Miroslav. Kontaktujte nás - PolovnaOptika.sk. *PolovnaOptika.sk* [online]. 2013 [cit. 2013-05-20]. Dostupné z: [http://polovnaoptika.sk/index.php?main\\_page=contact\\_us](http://polovnaoptika.sk/index.php?main_page=contact_us)
- 37) ZIKMUND, Martin. Kde se vzala a k čemu všemu je vlastně SWOT analýza. In: *BusinessVize.cz* [online]. 2010 [cit. 2013-05-20]. Dostupné z: <http://www.businessvize.cz/planovani/kde-se-vzala-a-k-cemu-vsemu-je-vlastne-swot-analyza>
- 38) KOCH, Miloš a Jan DOVRTĚL. *Management informačních systémů*. Vyd. 1. Brno: Akademické nakladatelství CERM, 2006, 174 s. ISBN 80-214-3262-4.
- 39) *Linux: dokumentační projekt*. 4., aktualiz. vyd. Brno: Computer Press, 2007, 1334 s. ISBN 978-80-251-1525-1.
- 40) About Yii framework. *Yii Framework* [online]. 2008 [cit. 2013-05-20]. Dostupné z: <http://www.yiiframework.com/about/>
- 41) BERNARD, Borek. Úvod do architektury MVC. In: *Zdroják* [online]. 2009 [cit. 2013-05-17]. Dostupné z: <http://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>

## SEZNAM OBRÁZKŮ

Obrázek 1: Marketingový mix 4C na internetu .....	15
Obrázek 2: Nákupní rozhodovací proces.....	18
Obrázek 3: Od hodnocení k rozhodnutí.....	19
Obrázek 4: Strategie dlouhého chvostu.....	24
Obrázek 5: Návrh klíčových slov v systému Sklik.cz.....	26
Obrázek 6: Výsledek v SERP Seznam.cz.....	27
Obrázek 7: Stránka v SERP Google.cz.....	27
Obrázek 8: Systém a okolí systému.....	35
Obrázek 9: Modelování systému.....	37
Obrázek 10: Elektronické obchodování.....	39
Obrázek 11: Princip fungování elektronického obchodu.....	41
Obrázek 12: Příklad registrace v e-shopu CZC.cz.....	42
Obrázek 13: Doplnková navigace a vyhledávání.....	44
Obrázek 14: Výchozí administrační rozhraní Magento.....	50
Obrázek 15: Administrační rozhraní Prestashopu.....	52
Obrázek 16: Hlavní části systému a uživatelské role.....	60
Obrázek 17: Use case diagram frontendu.....	61
Obrázek 18: Use case diagram backendu.....	62
Obrázek 19: Grafický popis procesu objenávky pomocí EPC diagramu.....	64
Obrázek 20: ER diagram produktového katalogu.....	66
Obrázek 21: ER diagram nákupního košíku.....	67
Obrázek 22: E-R Diagram objednávkové části.....	68
Obrázek 23: Návrhový vzor MVC.....	73
Obrázek 24: Kořenový adresář aplikace.....	73
Obrázek 25: Aplikační adresář.....	75
Obrázek 26: Vzhled naprogramovaného obchodu.....	79

## **SEZNAM TABULEK**

Časová kalkulace implantace open-source obchodu.....	53
Časové náklady realizace obchodu na míru.....	55
Silné stránky SWOT.....	57
Slabé stránky SWOT.....	57
Příležitosti SWOT.....	57
Hrozby SWOT.....	57

## **SEZNAM ZKRATEK**

<b>AJAX</b>	Asynchronous JavaScript and XML
<b>B2C</b>	Business to customer
<b>CSS</b>	Cascading Style Sheets
<b>DOM</b>	Document Object Model
<b>ERD</b>	Entity–relationship diagrams
<b>HTML</b>	HyperText Markup Language
<b>MVC</b>	Model View Controller
<b>OOP</b>	Objektově orientované programování
<b>PHP</b>	PHP Hypertext Preprocessor
<b>PPC</b>	Pay Per Click
<b>SEO</b>	Search engine optimization
<b>SERP</b>	Search engine results page
<b>SQL</b>	Structured Query Language
<b>UML</b>	Unified modeling language
<b>URL</b>	Uniform Resource Locator
<b>UTF-8</b>	UCS Transformation Format—8-bit

## **SEZNAM PŘÍLOH**

Příloha č.1 – Třída modelu košíku

Příloha č.2 – Controller objednávky

Příloha č.3 – View objednávky

Příloha č.4 – CD s instalací e-shopu, instalační pokyny

## PŘÍLOHA Č.1 – TŘÍDA MODELU KOŠÍKU

### <?php

```
/**
 * This is the model class for table "basket".
 *
 * The followings are the available columns in table 'basket':
 * @property integer $id_basket
 * @property integer $id_user
 * @property string $date_created
 * @property integer $isActiveBasket
 * @property integer $is_ordered
 *
 * The followings are the available model relations:
 * @property Product[] $products
 * @property Order[] $orders
 */
class Basket extends CActiveRecord
{

    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }

    public function tableName()
    {
        return 'basket';
    }

    /**

    public function rules()
    {
        return array(
            array('id_user', 'required'),
            array('id_user', 'numerical', 'integerOnly'=>true),
            array('date_created', 'safe'),
        );
    }
}
```

```

        array('id_basket, id_user, date_created', 'safe',
'on'=>'search'),
    );
}
public function relations()
{
    return array(
        'products' => array(self::MANY_MANY, 'Product',
'basket_item(id_basket, id_product)'),
        'orders' => array(self::HAS_ONE, 'Order', 'id_basket'),
    );
}

public function attributeLabels()
{
    return array(
        'id_basket' => 'Id Basket',
        'id_user' => 'Id User',
        'date_created' => 'Date Created',
    );
}

/**
 * Retrieves a list of models based on the current search/filter conditions.
 * @return CActiveDataProvider the data provider that can return the
models based on the search/filter conditions.
 */
public function search()
{
    $criteria=new CDbCriteria;
    $criteria->compare('id_basket',$this->id_basket);
    $criteria->compare('id_user',$this->id_user);
    $criteria->compare('date_created',$this->date_created,true);

    return new CActiveDataProvider($this, array(
        'criteria'=>$criteria,
    ));
}

```

```

public function beforeSave() {
    if ($this->isNewRecord){
        $this->date_created = new CDbExpression('NOW()');
    }
    return parent::beforeSave();
}

```

```

public function setActiveBasket(){
    $idUser=Yii::app()->user->id;
    $userBaskets=Basket::model()-
>findAllByAttributes(array('id_user'=>$idUser,'is_ordered'=>0));
    foreach($userBaskets as $item){
        $item->isActiveBasket=0;
        $item->save();
    }

    $this->isActiveBasket=1;
    $this->save();
    Yii::app()->user->setState('id_basket',$this->id_basket);
}

```

```

public function getTotalPrice(){
    $idBasket=$this->id_basket;
    $connection=Yii::app()->db;
    $sql='select SUM(count*price) as TotalPrice
        from basket_item left join product
        on basket_item.id_product=product.id_product
        where id_basket='.$idBasket;

    $command=$connection->createCommand($sql);
    $row=$command->queryRow();
    return round($row['TotalPrice'],2);
}

public function getItemCount(){
    $itemCount=0;
    $connection=Yii::app()->db;

```



```

        $sql='select SUM(count)as itemCount
                from basket_item
                where id_basket='.$this->id_basket;
        $command=$connection->createCommand($sql);
        $row=$command->queryAll();
        $itemCount=$row['0']['ItemCount'];
        if($itemCount===null){ $itemCount=0;}
        return $itemCount;

    }

    public function isEmpty(){
        if($this->getItemCount()==0){ return true;}
        else return false;

    }

    public function getWeight(){
        $command = Yii::app()->db->createCommand()
        ->select('SUM(count*weight)as weight')
        ->from('basket_item u')
        ->leftJoin('product p', 'u.id_product=p.id_product')
        ->where('u.id_basket=:id', array(':id'=>$this->id_basket))
        ->queryRow();

        return round($command['weight'],2);
    }

}

```

## PŘÍLOHA Č.2 – CONTROLLER OBJEDNÁVKY

```
<?php

class OrderController extends BasicController
{

    public function filters()
    {
        return array(
            'accessControl', // perform access control for CRUD operations
            'postOnly + delete', // we only allow deletion via POST request
        );
    }

    public function accessRules() {
        return array(

            array('allow',
                'actions'=>array('init','create','index', 'delete',
                    'update','view', 'selectCarrier', 'selectAddress', 'confirm', 'accepted'),
                'roles'=>array('customer'),

            ),

            array('deny',
                'users'=>array('*'),
            ),

        );
    }

    public function actionInit($id_basket){
        $basket=Basket::model()->findByPk((int)$id_basket);
        Yii::app()->session['order'] = array(
            'id_basket' => $id_basket,
            'weight'=>$basket->getWeight(),
            'id_user'=>$basket->id_user,
            'id_carrier'=>0,
            'carrier_price'=>0
        );
        $this->redirect(array('selectcarrier'));
        $this->render('init');
    }
}
```

```

public function actionSelectCarrier(){

    $weight=Yii::app()->session['order']['weight'];
    $avaiable=Carrier::model()->returnForWeight($weight);
    $methods=PaymentMethod::model()-
>findAllByAttributes(array('deleted'=>0,'is_active'=>1));

    if(isset($_POST['Carrier'])&&isset($_POST['PaymentMethod']))
    {
        $id_carrier=$_POST['Carrier'];
        $price=0;
        foreach($avaiable as $item){
            if($item['id_carrier']==$id_carrier){
                $price=$item['price'];
            }
        }
        $_SESSION['order']['id_carrier']=$id_carrier;
        $_SESSION['order']['carrier_price']=$price;
        $_SESSION['order']['id_payment_method']=$_POST['PaymentMethod'];
        $_SESSION['order']['payment_price']=
PaymentMethod::model()->findByPk($_POST['PaymentMethod'])->price;
        unset($_POST['Carrier']);
        unset($_POST['PaymentMethod']);
        $this->redirect(array('selectaddress'));

    }

    $this->render('selectcarrier',array('avaiable'=>$avaiable,
'methods'=>$methods));
}

public function actionSelectAddress(){
    $adresy=User::model()->findByPk(Yii::app()->user->id)-
>getUserAddresses('active');
    $_SESSION['back_select_address']='order/selectaddress';

    if(isset($_POST['AddressInvoice'])&&isset($_POST['AddressSend'])){
        $_SESSION['order']['id_address_invoice']=$_POST['AddressInvoice'];
        $_SESSION['order']['id_address_send']=$_POST['AddressSend'];
        unset($_SESSION['back_select_address']);
        $this->redirect(array('confirm'));
    }
}

```

```

    }

    $this->render('selectaddress',array('adresy'=>$adresy));
}

public function actionConfirm(){
    $order= Yii::app()->session['order'];

    $this->render('confirm',array('order'=>$order));
}

public function actionAccepted(){
    if(isset (Yii::app()->session['id_order'])){
        $id_order=Yii::app()->session['id_order'];
        $order=Order::model()->findByPk($id_order);
        $id_basket=$order->id_basket;
        Basket::model()->updateByPk($id_basket,
array('is_ordered'=>1,'isActiveBasket'=>0));
        $basket=new Basket();
        $basket->isActiveBasket=1;
        $basket->id_user=Yii::app()->user->id;
        $basket->setAsActiveBasket();
        if($basket->save()){
            $_SESSION['id_basket']=$basket->id_basket;
        }
        unset(Yii::app()->session['id_order']);
        $this->render('accepted',array('id_order'=>$id_order));
    }
    else $this->redirect (array('basket/index'));
}

```

```

public function actionCreate()
{
    $model=new Order('Create');
    $model->id_basket =(int)Yii::app()->session['order']['id_basket'];
    $model->weight = (int)Yii::app()->session['order']['weight'];
    $model->id_user=(int)Yii::app()->session['order']['id_user'];
    $model->id_carrier=(int)Yii::app()->session['order']['id_carrier'];
    $model->carrier_price=(int)Yii::app()->session['order']['carrier_price'];
    $model->payment_method_price=(int)Yii::app()->session['order']['payment_price'];
}

```

```

$model->id_payment_method=
(int)Yii::app()->session['order']['id_payment_method'];
$model->id_address_invoice=(int)Yii::app()->session['order']['id_address_invoice'];
$model->id_address_send=(int)Yii::app()->session['order']['id_address_send'];
$total_price=$model->payment_method_price+$model->carrier_price
    +Basket::model()->findByPk($model->id_basket)->getTotalPrice();
$model->total_price=(int)$total_price;
echo $model->errors;
if($model->save()){
    $_SESSION['id_order']=$model->id_order;
    unset(Yii::app()->session['order']);
    $this->redirect(array('accepted'));
    $this->render('create',array(
        'model'=>$model,
    ));
}
public function actionView($id)
{
    $this->render('view',array(
        'model'=>$this->loadModel($id),
    ));
}
public function actionUpdate($id)
{
    $model=$this->loadModel($id);

    if(isset($_POST['Order']))
    {
        $model->attributes=$_POST['Order'];
        if($model->save())
        $this->redirect(array('view','id'=>$model->id_order));
    }

    $this->render('update',array(
        'model'=>$model,
    ));
}

public function actionDelete($id)
{
    $this->loadModel($id)->delete();
}

```

```

        if(!isset($_GET['ajax']))
            $this->redirect(isset($_POST['returnUrl']) ? $_POST['returnUrl'] :
array('admin'));
    }

    public function actionIndex()
    {
        $idUser=Yii::app()->user->id;
        $dataProvider=new CActiveDataProvider('Order', array(
            'criteria'=>array(
                'condition'=>'id_user='.$idUser,
                'order'=>'date_created DESC',

            ),
            'pagination'=>array(
                'pageSize'=>20,
            ),
        ));
        $this->render('index',array('dataProvider'=>$dataProvider
        ));
    }

    public function actionAdmin()
    {
        $model=new Order('search');
        $model->unsetAttributes(); // clear any default values
        if(isset($_GET['Order']))
            $model->attributes=$_GET['Order'];

        $this->render('admin',array(
            'model'=>$model,
        ));
    }

    public function loadModel($id)
    {
        $model=Order::model()->findByPk($id);
        if($model===null)
            throw new CHttpException(404,'The requested page does not
exist.');
```

return \$model;

```
}

protected function performAjaxValidation($model)
{
    if(isset($_POST['ajax']) && $_POST['ajax']==='order-form')
    {
        echo CActiveForm::validate($model);
        Yii::app()->end();
    }
}
```

## PŘÍLOHA Č.3 – VIEW OBJEDNÁVKY

<?php

```
/* @var $this OrderController */
```

```
/* @var $dataProvider CActiveDataProvider */
```

?>

<h1>Objednávky</h1>

<?php

```
$this->widget('zii.widgets.grid.CGridView', array(
    'dataProvider'=>$dataProvider,
    'columns'=>array(
        array(
            'header'=>'Vytvořena',
            'name'=>'date_created',
        ),
        array(
            'header'=>'Číslo objednávky',
            'name'=>'id_order',
        ),
        array(
            'header'=>'Fakturační adresa',
            'name'=>'id_address_invoice',
            'value'=>'CHtml::encode(Address::model()->findByPk($data-
>id_address_invoice)->returnAsRow())'
        ),
        array(
            'header'=>'Dodací adresa',
            'name'=>'id_address_send',
            'value'=>'CHtml::encode(Address::model()->findByPk($data-
>id_address_send)->returnAsRow())'
        ),
        array(
            'header'=>'Dodací adresa',
            'name'=>'idAddressSend.name',
        ),
        array(
            'header'=>'Cena dopravy [EUR]',
            'name'=>'carrier_price'
        ),
    ),
```



```
array(  
  'header'=>'Dopravce',  
  'name'=>'idCarrier.name'),  
  array(  
    'header'=>'Cena za platbu [EUR]',  
    'name'=>'payment_method_price'  
  ),  
  array(  
    'header'=>'Platební metoda',  
    'name'=>'idPaymentMethod.name'),  
  array(  
    'header'=>'Celková cena objednávky [EUR]',  
    'name'=>'total_price'  
  ),  
  array(  
    'header'=>'Vyřízena',  
    'name'=>'expedited'  
  ),  
),  
));  
?>
```